DOE/NASA/0017-3
NASA CR-174721

Manual of Phosphoric Acid Fuel Cell Power Plant Optimization Model and Computer Program

Cheng-yi Lu and Kalil A. Alkasab
Cleveland State University

May 1984

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Under Grant NCC 3-17

for
U.S. DEPARTMENT OF ENERGY
Morgantown Energy Technology Center

DOE/NASA/0017-3
NASA CR-174721

Manual of Phosphoric Acid Fuel Cell Power Plant Optimization Model and Computer Program

Cheng-yi Lu and Kalil A. Alkasab
Cleveland State University
Cleveland, Ohio 44115

May 1984

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, Ohio 44135
Under Grant NCC 3-17

for
U.S. DEPARTMENT OF ENERGY
Morgantown Energy Technology Center
Morgantown, West Virginia 26505
Under Interagency Agreement DE-AI21-80ET17088

TABLE OF CONTENTS

INTRODUCTION

An optimized cost and performance model for a phosphoric acid fuel cell powerplant system has been derived and developed into a modular FORTRAN computer code. The mathematical model developed in this study combines appropriate cost, energy, and electrochemical analysis as applied to the major components of a phosphoric acid fuel cell system.

In the model, the following important parameters were optimized: steam to methane ratio in the reformer, hydrogen utilization in the fuel cell stack, and the number of fuel cell plates per stack.

The optimization technique used here was by means of COMPUTE code which was developed for soliving nonlinear programming problems.

I. SYSTEM DESCRIPTION

As shown in Figure 1, methane which is circulated by compressor (C) is preheated by heat exchanger E-1 prior to mixing it with the super heated steam which receives its heat by passing through heat exchanger E-9. Before entering the reformer, the methane steam mixture is heated via heat exchangers E-2 and E-3. Inside the reformer, methane is catalytically reformed by reaction with excess steam to produce carbon monoxide, carbon dioxide, and the desired product, hydrogen. The effluent from the reformer is cooled by flowing through heat exchanger E-2 before it enters the high temperature shift converter S-1. The function of the high temperature shift converter is to increase the hydrogen concentration and to reduce the carbon monoxide concentration of the reformer gas effluent. The temperature of the effluent from the shift converter S-1 is then reduced by passing through heat excangers E-1, E-9 and E-6 before entering the low temperature shift converter S-2. The low temperature shift converter further increases the hydrogen concentration by promoting the shift reaction at a lower operating temperature. The effluent from the low temperature shift converter then enters the fuel cell containing H_2, CO, CH_4, CO_2 and H_2O. The fuel cell converts inputs of hydrogen and oxygen to DC power, water and heat. Oxygen is delivered to the fuel cell by air compressor A, which also provides air to the reformer burner. The spent fuel from the fuel cell anode goes to the burner after mixing with air supplied by compressor A.

Before entering the burner, the mixture is preheated by the burner effluent via heat exchanger E-4. The spent fuel is then burned with whatever additional methane is needed to provide the thermal energy necessary for the reformer reaction.

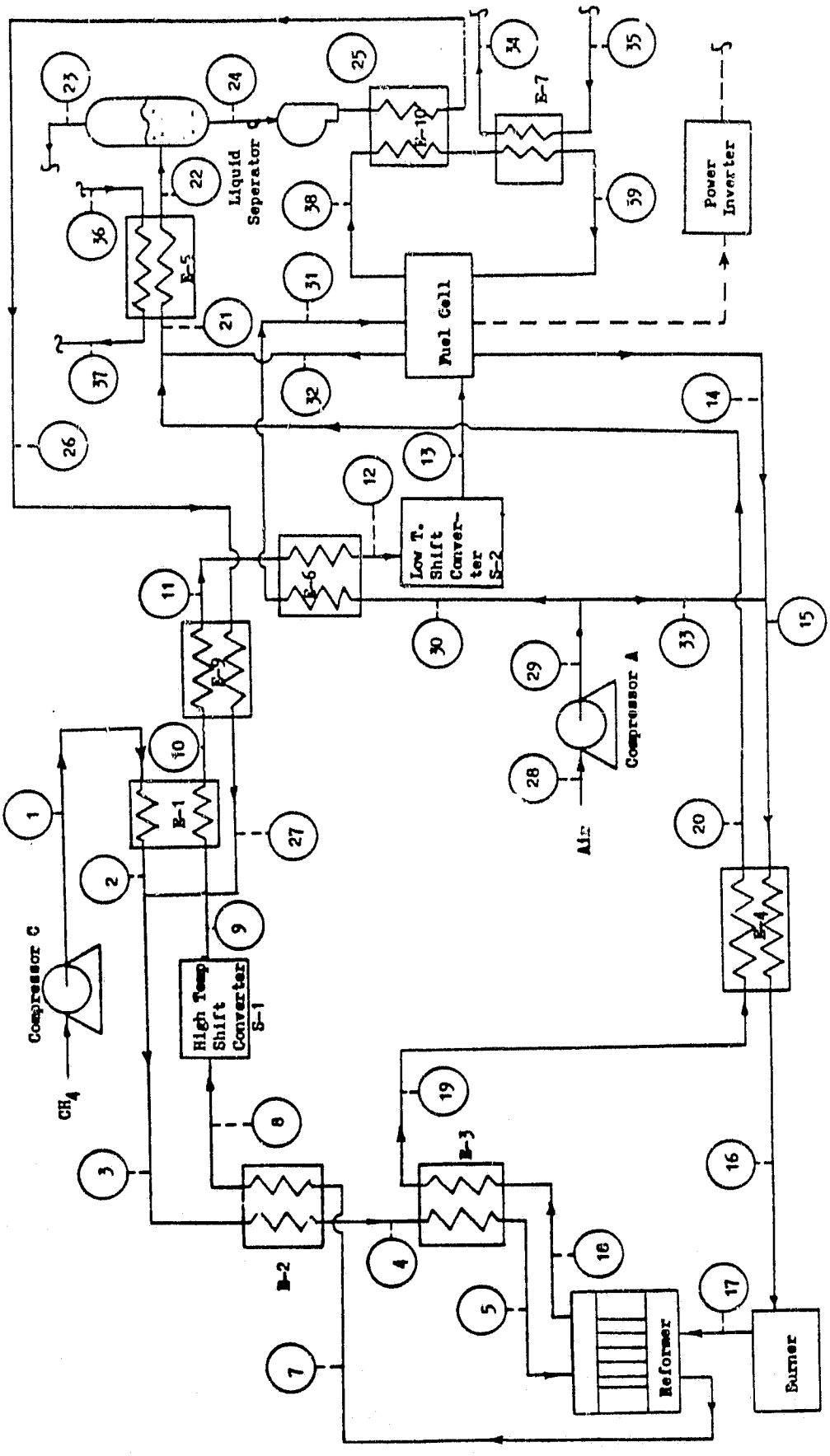

Figure 1 Flow diagram of GSU designed PAFC system

Heat generated in the fuel cell is removed by heat exchangers E-7 and E-10. Heat from heat exchanger E-7 can then be utilized in industrial heat processing or space heating and cooling, while exchanger E-10 is used to preheat the water supplied by liquid separator Q to provide the necessary steam needed for the reforming process. The effluents from the burner and fuel cell cathode will have their water removed and separated by condenser E-5 and liquid separator Q before allowing them to be exhausted to the atmosphere.

II. OPTIMIZATION MATHEMATICAL MODEL

This chapter describes a practical procedure for the optimization in designing a processing system like the 100 kW phosphoric acid fuel cell (PAFC) powerplant shown in Figure 1. The design criteria are the investment cost and the operation cost, both to be minimized.

Computation time increase rapidly with the number of variables to be optimized. The optimization procedure described here will, for brevity, be limited to three of the most important variables. Therefore, we will describe the procedure for the determination of the optimum design of hydrogen utilization in the stack, the steam to carbon ratio in the reformer, and the number of cell plates per stack. The derivation of the optimization model was based on the following assumptions.

1. All reactants and products are assumed to be ideal gases.

2. All heat exchangers in the PAFC system are assumed to be of the counter flow type.

3. In the reformer and shift converters, all processes are assumed to be isothermal. In addition, only the demethanation and water shift reactions (Ref. 1) occur in the reformer, and the latter reaction reaches the equilibrium condition.

4. Outlet temperature of the fuel cell stack is equal to the stack operating temperature.

5. The combustion process in the burner is assumed to go to completion.

To carry out the required optimization, performance and cost analyses of the PAFC system will be essential. Energy, mass, and electrochemical analyses in the reformer, the heat exchangers, the shift converters, and the fuel cell stack were combined to develop a mathematical model and the associated computer program for PAFC powerplant (Ref. 1).

In the following cost analysis and formulation of the optimization problem will be presented.

2.1 Cost Analysis

The cost, which is to be minimized, is the sum of the equipment capital costs. For the sake of simplicity, it will be assumed that the cost of piping and auxiliary equipment (like pump, compressors, and separator) are negligible.

To minimize equipment cost, expressions for the installed cost for each primary equipment are required. Such expressions are readily obtained by using the exponent rule for cost estimation (Ref. 2). The rule expresses the investment of each unit equipment to be a function of paraments that can be evaluated in terms of computed quantities and decision variables. Based on mid-1981 money, the following are installed cost expressions for the reformer: Cr, high and low temperature shift converters: Chs and Cls, respectively, fuel cell module; Cf, and heat exchangers: Ce.

$$Cr = 309.82\ W^{0.63} \quad (1)$$

$$Chs = 162.65\ (H7+H9)^{0.69} \quad (2)$$

$$Cls = 230.34\ (H9 + H13)^{0.69} \quad (3)$$

$$Cf = 8.5136\ (N)(Nst) \quad (4)$$

$$Ce = 1046.23\ (A1^{0.6924} + A2^{0.6924} + A3^{0.6924} + A4^{0.6924} + A6^{0.6924} + A9^{0.6924}) \quad (5)$$

where W : amount of catalyst in the reformer, lb

Hi : flow rate of steam i, lb-mole/hr

N : number of fuel cell plates in one stack

Nst: number of stacks in the system

Aj : transfer area of heat exchanger j, m

The cost of the power inverter can be obtained from interpolating the following cost table (Ref. 3).

Voltage-V:	50	164	203	248	304	366	433	528	657
Cost-$/V:	200	160	150	140	130	120	110	100	90

2.2 Formulation of the Optimization Problem

In designing the PAFC powerplant, a number of constraints were imposed on its construction and operation. The following is description of these limitations:

1. Since carbon monoxide is considered as poison for the fuel cell catalyst, a limit of 1 percent volume in the fuel cell feed was placed. Thus, the first optimization contraint is:

$$g1 = Yco \leq 0.01 \tag{6}$$

where Yco is the mean mole fraction of carbon monoxide.

2. Because platinum and carbon components corrode at cell potentials above 0.8 V, and since lower potential means lower cell efficiency, the cell operating voltage is to be limited to the range of 0.5 to 0.75V. This voltage range provides the following constraints:

$$g2 = V \leq 0.75 \tag{7}$$

$$g3 = V \leq -0.5 \tag{8}$$

3. For small fuel cell module consideration, it is assumed that the maximum output voltage per stack not to exceed 300 V. This provides the fourth constraint:

$$g4 = Vs \leq 300 \tag{9}$$

where Vs is the output voltage per stack.

4. The system is designed for 100 kW electric output. This provides a fifth constraint that can be expressed as,

$$g5 = \left\{1.0148 - [1.0298-0.001689\ (5.0976-DC/1000)]^{1/2}\right\} /0.0008444 \leq 100 \tag{10}$$

where DC is the DC electric power output.

5. Since the electrolyte, phosphoric acid, evaporates significantly at temperature above 493K, and because lower operating temperature results in lower cell efficiency, the inlet temperature of the fuel entering from the

anode side and the process air entering from the cathode side is to be limited to the range of 398K to 477K. This provides the following constraints:

$$G6 = T12 \leq 477 \tag{11}$$

$$G7 = T12 \leq -398 \tag{12}$$

$$G8 = T31 \leq 477 \tag{13}$$

$$G8 = T31 \leq -398 \tag{14}$$

where subscripts 12 and 13 refer to the stream numbers 12 and 13 in Figure 1.

6. To provide high conversion in fuel processing subsystem, it is assumed that the concentration of hydrogen entering the cell anode is greater than 60 percent. This provides the following constraint,

$$G10 = -H13/F13 \leq -0.6 \tag{15}$$

where, H and F refer to amount of hydrogen and fuel, respectively, and subscript 13 refer to the steam number 13 in Figure 1.

With the formulation of the constraints imposed on the system design being completed, the optimization problem can be stated as

$$\text{Minimum } \sum_i Ci = Cr + Chs + Cls + Cf + Ce + Cv \tag{16}$$

$$\text{Subject to: } g1, g2, \ldots, g10 \text{ (Equations 6 to 15)} \tag{17}$$

this is a constrained minimizing problem with three decision variables and ten constraints.

2.3 Optimization Algorithm

The constrained optimization problem (equations 16 and 17) was solved by using mixed penalty function together with unconstrained optimization technique for extremization. This nonlinear constrained problem,

$$\text{Min. } \phi = \sum_{i} Ci + K \sum_{j} \ln(g_j) \qquad (18)$$

where, K is the penalty coefficient and ϕ is a new objective function.

A Fortran computer program had been developed to solve the optimization problem (Ref. 4) called COMPUTE, which will be described in the next chapter. Values of ϕ were minimized for a sequence of monotonically increasing $K > 0$. Then, by means of Hooke and Jeeve pattern search method (Ref. 5), the optimal point can be determined from an assigned input for different values of the penalty function.

III. OPTIMIZATION COMPUTER MODEL

The MAIN program calls three subroutines which are PATTRN, OBJECT, and COEFF. Subroutine OBJECT calls four subroutines which are DMIX1, SCALE, RLIN, and HEXC1, where DMIX1 and HEXC1 are the same as those used in the performance model and RLIN is also used in the cost model. Subroutine CONV1 is called by Subroutine DMIX1 for solving a nonlinear algebraic equation. The overall structure of the program for evaluating some optimal design values of PAFC powerplant is illustrated in Figure 2.

3.1 General Optimization Program - COMPUTE

The COMPUTE code (Ref. 5) was used as the general optimization program. COMPUTE solves the constrained optimization problems using mixed penalty function* together with unconstrained optimization techniques for extremization. There are three optimal search techniques which can solve our optimization problems with the mixed penalty function. These are: Hooke and Jeeves pattern search (Ref. 4), Fletcher and Reeves conjugate-gradient search (Ref. 6), and Fletcher and Powell's variable-metric search (Ref. 6). The Hooke and Jeeves pattern search method was selected. It performs one search from an input starting point. To save computation time, we also fixed the initial penalty level instead of using Subroutine COEFF to find one.

---

*By mixed penalty functions, we mean if the first ℓ constraints are inequalities, and constraints $(\ell+1)$ to m are equalities, our problem becomes:

$$\min. \quad \phi = f_0(\underset{\sim}{x}) + K_j \left[\sum_{i=1}^{\ell} \sigma_i (g_i(\underset{\sim}{x})) + \sum_{i=\ell+1}^{m} (g_i(\underset{\sim}{x}))^2 \right]$$

The function $\phi(\underset{\sim}{x}, K)$ is then minimized for a sequence of monotonically increasing $K > 0$.

Figure 2 Overall Structure of Optimization Computer Model

There are one Main program and three Subroutines adopted from COMPUTE code to solve this optimization mathematical model (Equations 16 and 17). These will be discussed in the following sections.

MAIN Program

The MAIN program reads the input data and checks whether the initial guessed data in the limits of the assigned bounds, then controls the procedures for the convergence of the penalty function. The nonlinear constrained problem is first changed to the unconstrained problem by setting up the penalty function. Then, by means of pattern search technique, one can find the optimal point at different penalty levels. For a series of increasing values of penalty coefficient (or penalty level), the sequence of resulting solutions will more and more closely satisfy the constraints. When all the constraints are satisfied within some preassigned tolerance, the procedure terminates.

The nomenclature of MAIN program is shown in Table 1, while the flow chart of this program is shown in Figure 3.

Subroutine OBJECT

This subroutine computes the modified objective function. Constraints and objective function defined beforehand are calculated in this subroutine. The trial and error procedure for simulating the continuous PAFC system is the same as those in the performance model, and will not repeat here. The cost estimated in the objective function is shown in Equation 16, and the constraints are Equations 6 to 15.

TABLE 1

NOMENCLATURE IN MAIN PROGRAM AND SUBROUTINE OBJECT

N:	number of independent variables
JEQ:	number of functional equality constraints
JINEQ:	number of functional inequality constraints
M:	number of functional constraints
IPMAX:	maximum number of penalty levels
IOBJ:	function evaluation counter
IP:	penalty level counter
IR:	random check improvement counter
IT:	search iteration counter
ITMAX:	maximum number of iterations per gradient search
ITOBJ:	total number of iterations per gradient search
ITOT:	total number of search iterations
NTEST:	number of randomly generated exploratory points in the pattern search
ALPHA:	scaling factor between SIGMA2 and SIGMA1
ARAND:	initial parameter for the random number generator
CFAC:	proportionally constant used in calculating PNLTY
CONST:	penalty multiplier
CONVG:	penalty level convergence criterion
DELTA:	finite-difference increment
DX:	length of a gradient move
EFAC:	search increment reduction factor
EPSLN:	gradient vector convergence factor
GRADX:	gradient vector convergence criterion

TABLE 1 (cont'd)

NOMENCLATURE IN MAIN PROGRAM AND SUBROUTINE OBJECT

THETA: maximum time allowed to evaluate the objective function when calculating the gradient vector by central differences

GB(I): unscaled value of constraint I

G(I): scaled value of constraint I

SC(I): scaling factor for constraint I automatically determined at the beginning of each penalty level: G(I)=GB(I)*SC(I)

PD(I): sum of the magnitudes of D(GB(I))/(D(X(KFR)), KFR=1, 2, ..., N

SIGMAO: sum of the squares of the constraints which are violated (excluding bounds on the independent variables)

SIGMA1: sum of the squares of the scaled constraints which are violated (excluding the independent variable bounds)

SIGMA2: penalty function composed of the upper and lower bounds on the independent variables

SIGMA: sum of SIGMAO and a modified SIGMA2 obtained by dividing SIGMA2 by the penalty coefficeint squared

TMTOT: total search time

Program Options

IOUT=0 output follows final optimization

IOUT=1 output follows optimization for each starting point

IOUT=2 output follows optimization for each penalty level

IOUT=3 output follows each major search or gradient move

KBOUND=0 independent variables are unbounded

KBOUND=1 independent variables are non-negative

KBOUND=2 independent variables are bounded by specific values

KCLIMB=0 modified objective function optimized via Hooke and Jeeves pattern search

TABLE 1 (cont'd)

NOMENCLATURE IN MAIN PROGRAM AND SUBROUTINE OBJECT

KGRID=0 program uses fine-grid hillclimbing convergence criterion at all penalty levels

KGRID>0 program uses fine-grid hillclimbing convergence criterion at early penalty levels

KKOFF=0 automatic termination if Z not uniquely determined by X-vector

KKOFF>0 computation continues if Z not uniquely determined by X-vector

KMIN=0 program maximizing the objective function

KMIN=1 program minimizing the objective function

KSCALE=0 constraints scaled automatically

KSCALE=1 constraints not scaled

KSCALE=2 constraints not scaled during a penalty level due to detection of suspected constraint ridge by automatic testing procedure

KSENS=0 final values of partial derivatives are not calculated

KSENS>0 partial derivatives of objective function are calculated after optimization has been completed

NCHECK=0 program terminates without a final random check

NCHECK>0 program generates NCHECK thousand random check points after the optimum

NRAND=0 program performs one search from specified starting point

NRAND>0 program performs NRAND searches from specified starting point

Definition of Variables Used in OBJECT

AREA: area of fuel cell, cm^2/cell

C11, C12, C13, C14: coefficient used in calculating equation convergence of water shift reaction

CD: current density, A/cm^2

CN1: Q*A/CMIN in heat exchanger

TABLE 1 (cont'd)

NOMENCLATURE IN MAIN PROGRAM AND SUBROUTINE OBJECT

CK1, CK2:	coefficient used in calculating kinetic equation of demethanation
EXT:	extra air in burner
NST:	number of stacks in fuel cell
PH2O:	mole fraction of H_2O
PO2:	mole fraction of O_2
TAT:	ambient temperature, K
TOPFC:	operating temperature of fuel cell, K
U1:	total heat transfer coefficient of heat exchangers (assume the same U value for all exchangers and condenser), cal./m^2-hr-c

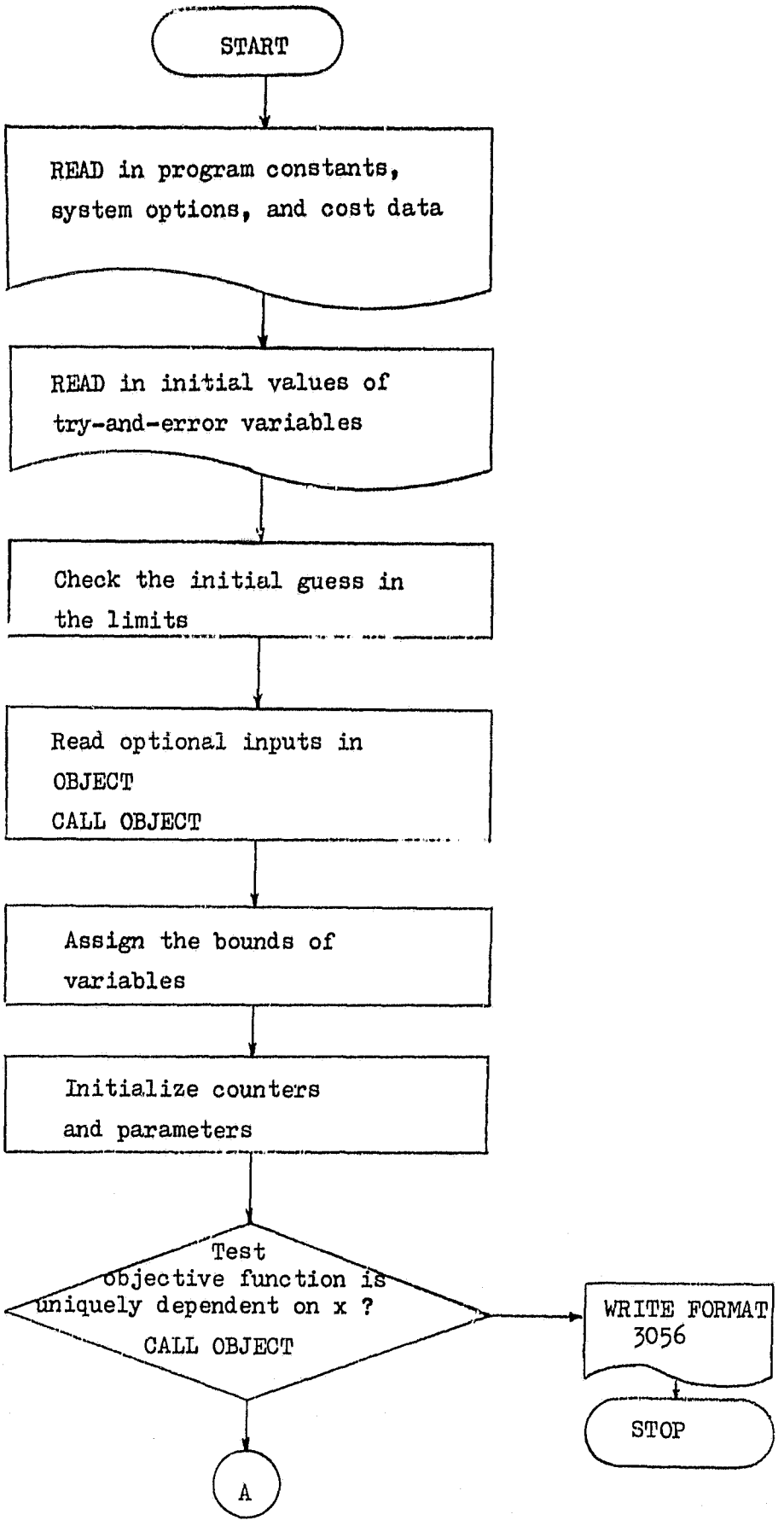

Figure 3 Flow Chart of Main Program

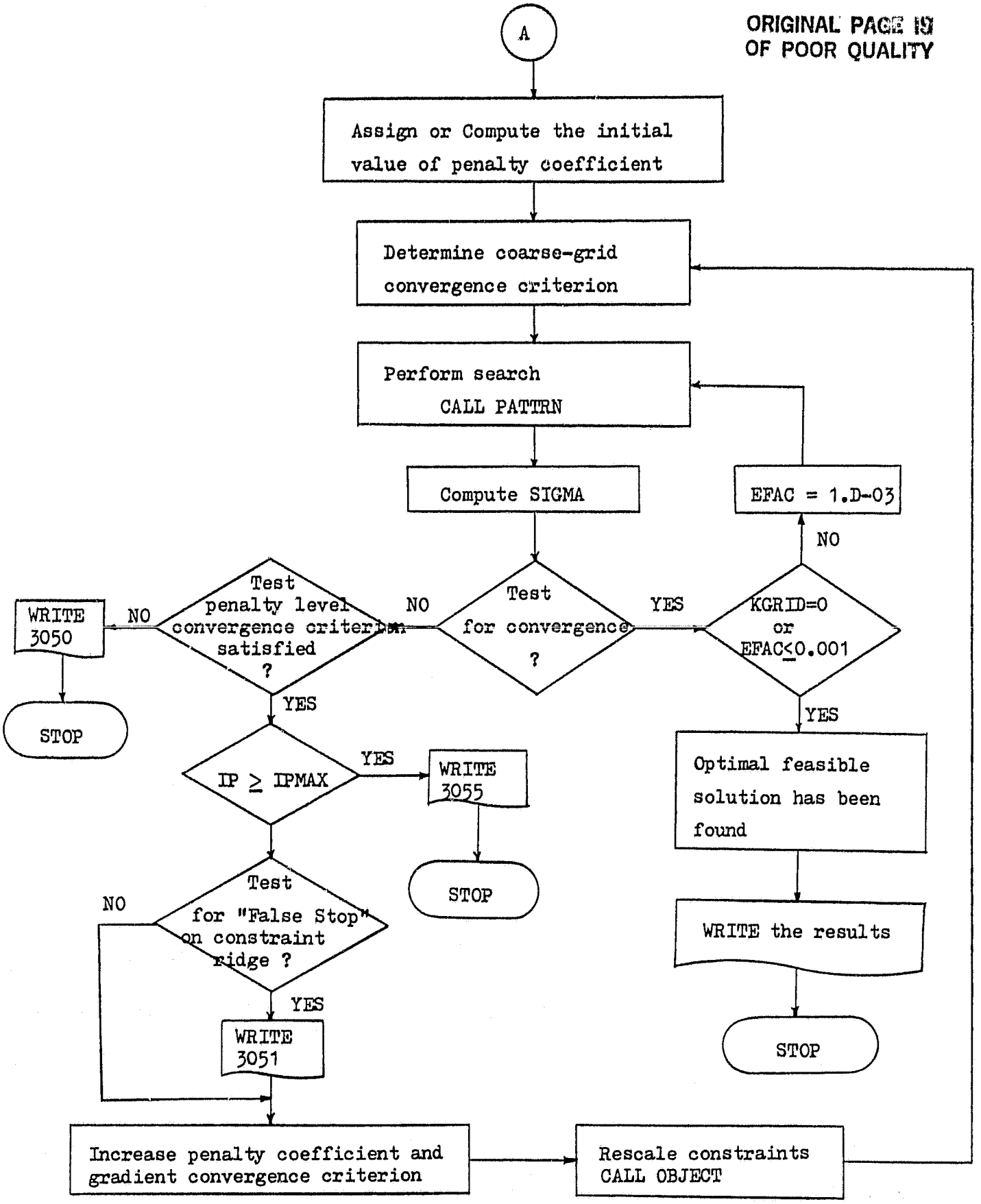

Figure 3 continued

The nomenclature of this subroutine is listed in Table 1. The program control variables are as follows:

K=1: read optional input and return

K=2: evaluate objective function and return

K=3: evaluate objective function, write optional output and return

K=4: scale constraints, evaluate objective function, and return

K=5: write optional output and return

K=6: evaluate objective function and unscaled SIGMA and return

ISCALE=1: constraints as yet unscaled

ISCALE=2: constraints already scaled

ISCALE=3: constraints scaling in progress

Subroutine PATTRN

This subroutine uses the Hooke and Jeeves algorithm (Ref. 4), known as pattern search, to do multidimensional search for the better point. This technique alternates sequences of local exploratory moves with extrapolations. The basis for the method is the intuitive presumption that a strategy which was successful in the past will be successful in the future.

The computation begins by selecting an initial exploratory point x_1^*, and evaluating the objective function $y(x_1^*)$. Then we increase x_1 to $x_1 + E_1$ and evaluate y at this new point. If y shows improvement as a result of this move, then we let this new point be the next exploratory point, x_{11}^*. If y does not improve at $x_1 + E_1$, we then try $x_1 - E_1$. In the event that y does not show improvement at both cases, we set x_{11}^* equal to the initial exploratory point x_1^*

From x_{11}^* we proceed to change x_2 by $\pm E_2$ and, thus establish the position of the next exploratory point x_{12}^*, as above. The procedure is repeated until all the independent variables have been perturbed by some distances. We shall refer to this terminal point as a base point, x_1.

A pattern move is now made,

$$x_2^* = x_1^* + 2\,(x_1 - x_1^*) \tag{19}$$

Another sequence of exploratory moves is then carried out. The best value of y found during this exploratory process is compared with the previous base point. If it is an improvement, it is accepted as new base point.

The next pattern move is

$$x_3^* = x_1^* + 2\,(x_2 - x_1^*) \tag{20}$$

and after that each pattern move (except the first) is made in accordance with the expression

$$x_{i+1}^* = x_{i-1} + 2\,(x_i - x_{i-1}) \tag{21}$$

In case that a pattern move is not successful, then x_{i+1}^* is set equal to x_i so that a new exploratory sequence is initiated from the previous base point. The successive explore-and-extrapolate strategy is again applied as before, providing that the objective function again continues to improve.

The flow chart of this subroutine is shown in Figure 4.

Subroutine SCALE

This subroutine computes the multiplicative scaling constants SC(I) such that the sums of the magnitudes of the partials of the constraints are roughly equal. The constraints in any one problem are often concerned with quantities of entirely different magnitudes. In one constraint, one may be dealing with values of the magnitude of 10^6, and in another with quantities having a magnitude of 1. In this form, change in the former constraint will completely mask any changes in the latter. When this situation arises, meaningful results cannot be obtained unless one "scales" all constraints. This subroutine uses the Keefer and Gottfried (Ref. 7) method to scale constraints.

The flow chart of this subroutine is shown in Figure 5.

3.2 Program Operation

The only program input consists of NAMELIST datasets which are read in a specified order. The first NAMELIST set is called CON, and contains the general information of the optimization procedure and the information of the specified problem to be solved. The order of input data inside one NAMELIST is unimportant.

The second set, INIT, contains the numerical values for the flow rate of the input fuel, the ambient temperature, the operating conditions of heat exchangers, and the operating conditions in the fuel cell stacks.

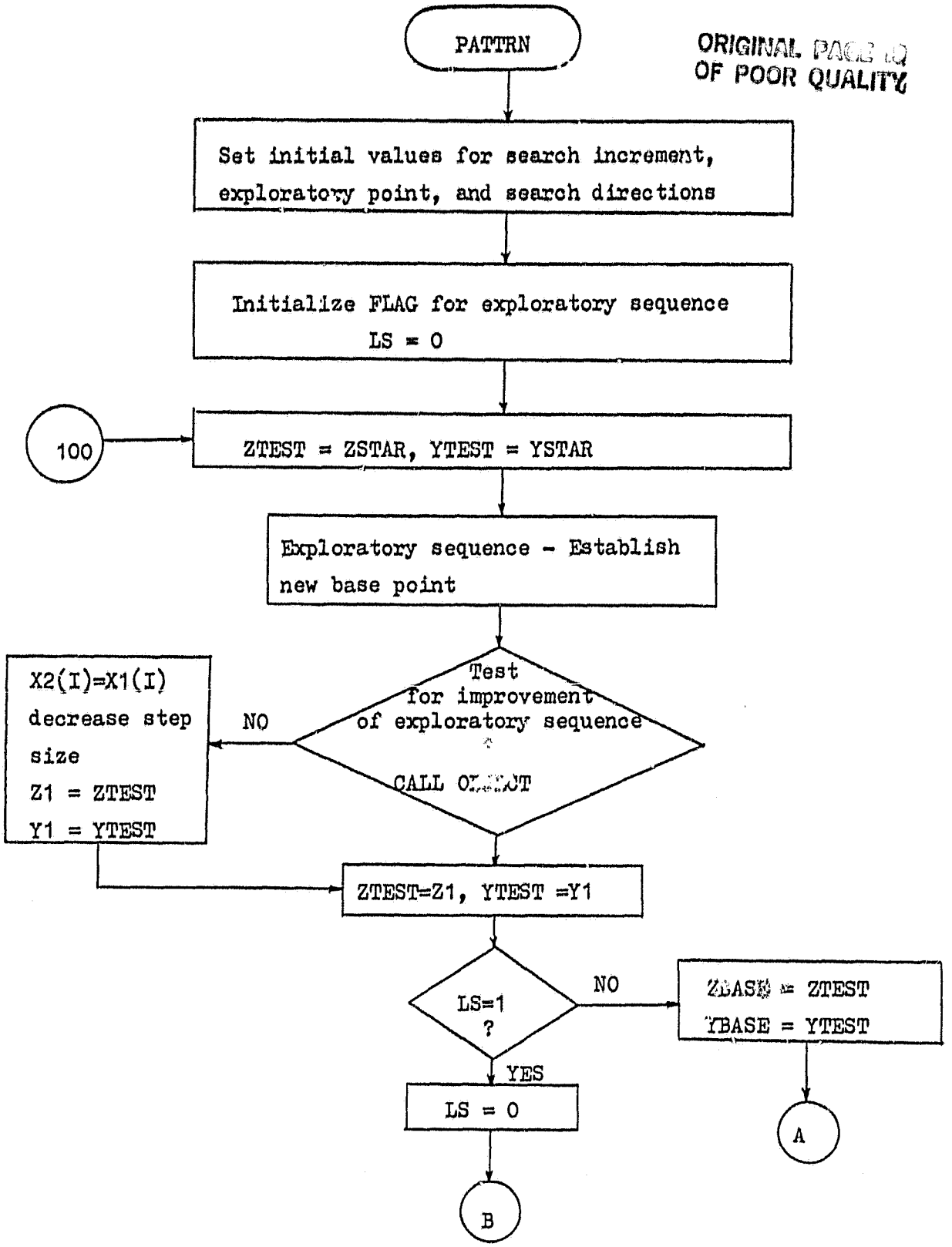

Figure 4 Flow Chart of Subroutine PATTRN

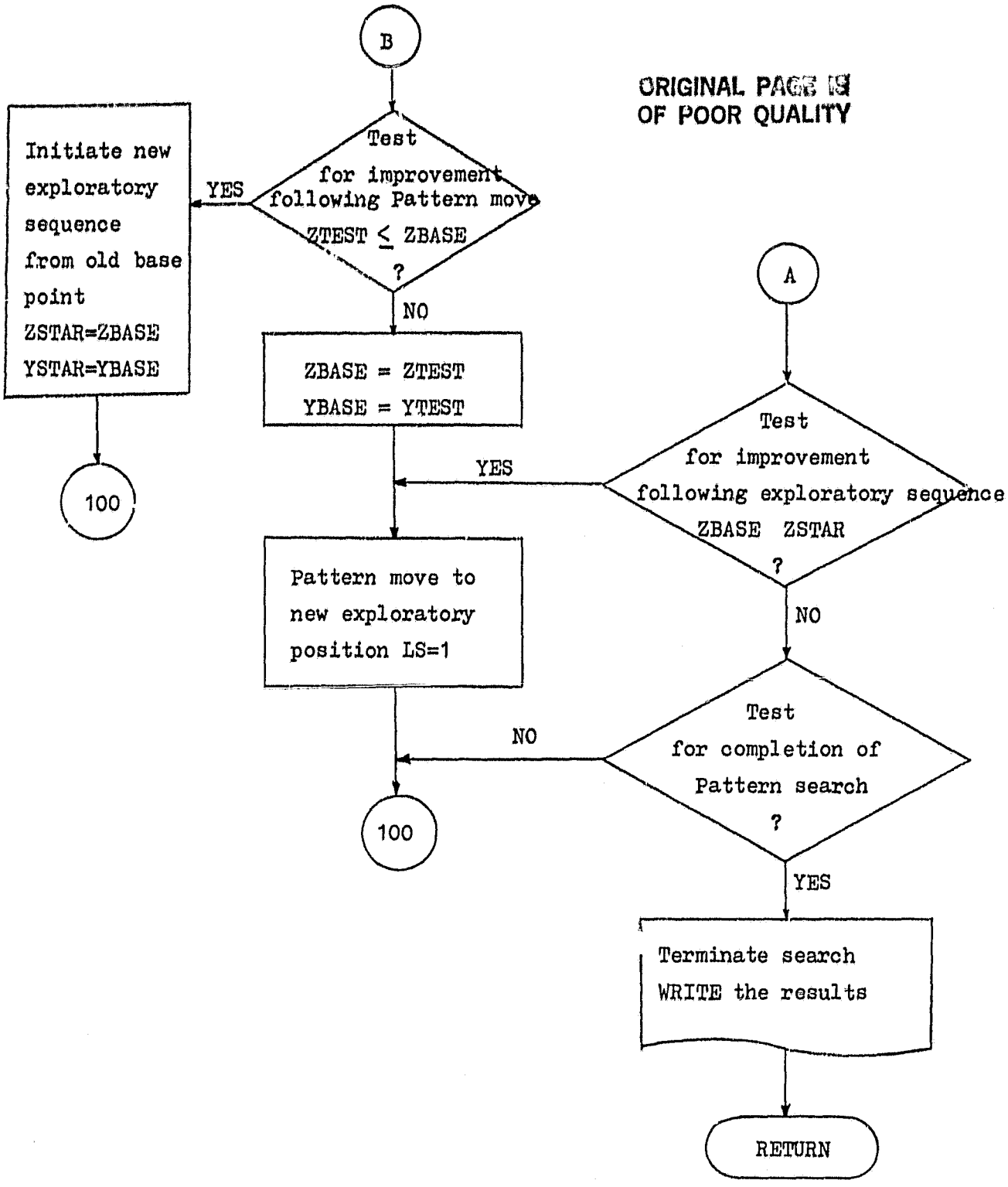

Figure 4 continued

The third set, CON1, carries the constants used in evaluating the reaction rate of demethanation reaction and the equilibrium constant of the water shift reaction in the fuel processor.

The fourth set, CON2, contains the coefficients used in estimating the cost of the power inverter.

The last NAMELIST dataset, S1, gives the initial guesses for the flow rates and temperatures of specified flowsheet streams (Figure 1). In addition, the bounds and initial guesses of decision variables are also listed in S1.

More detailed descriptions of these input data are shown in Table 2, along with their dimensional units, and the numerical values used in the sample run.

Besides the input data, the values of some variables have been fixed in the program as the initial assumption, which is listed in Table 3.

3.3 Sample Run

The object of the sample problem is to calculate the optimal values of hydrogen utilization (x_1) in the fuel cell stack, the steam to carbon ratio (S/C, x_2) in the reformer, and the number of cell plates per stack (x_3) in order to minimize equipment capital cost of the 100 kW (AC) PAFC powerplant system. The objective function and constraints are listed in Chapter II. The input data for this sample problem is shown in Table 2. To avoid carbon formation inside the reformer, the minimal value of x_2 was assigned a higher

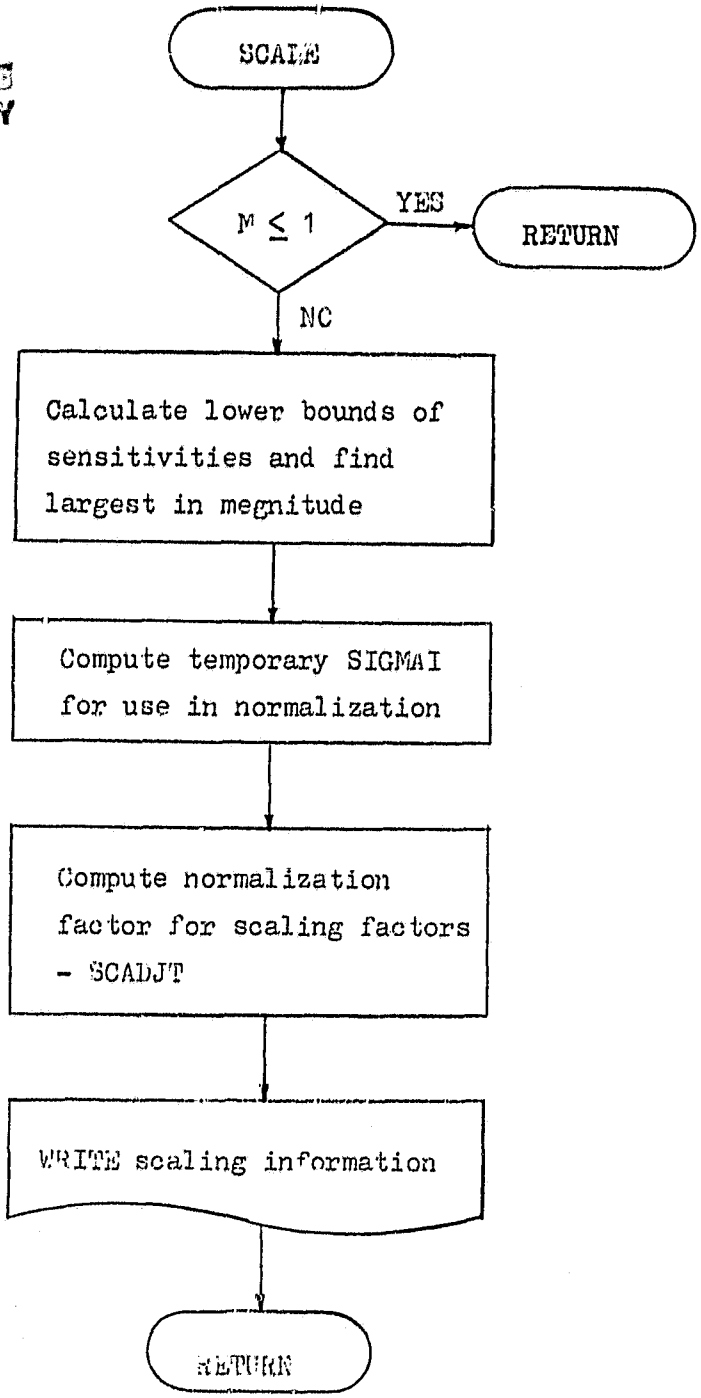

Figure 5 Flow Chart of Subroutine SCALE

TABLE 2

INPUT DATA FOR OPTIMIZATION COMPUTER MODEL (SAMPLE RUN)

N:	number of independent variables	3
JEQ:	number of functional equality constraints	0
JINEQ:	number of functional inequality constraints	10
IPMAX:	maximum number of penalty levels	25
ALPHA:	scaling factor between SIGMA2 and SIGMA1	1.
ARAND:	initial parameter for the random number generator	0.314159167
CFAC:	proportionally constant used in calculating PNLTY	0.01
CONST:	penalty multiplier	8.
CONVG:	penalty level convergence criterion	0.0001
DELTA:	finite-difference increment	1.D-10
DX:	length of a gradient move	10.
EFAC:	search increment reduction factor	0.001
EPSLN:	gradient vector convergence factor	1.D-05
GRADX:	gradient vector convergence criterion	0.
THETA:	maximum time allowed to evaluate the objective function when calculating the gradient vector by central differences	0.1
IOUT=0	output follows final optimization	
KBOUND=2	independent variables are bounded by specific values	
KCLIMB=0	modified objective function optimized via Hooke and Jeeves pattern search	
KGRID=1	program uses fine-grid hillclimbing convergence criterion at early penalty levels	
KKOFF=0	automatic termination if Z not uniquely determined by X-vector	
KMIN=1	program minimizing the objective function	

TABLE 2 (cont'd)

INPUT DATA FOR OPTIMIZATION COMPUTER MODEL (SAMPLE RUN)

KSCALE=0	constraints scaled automatically	
KSENS=0	final values of partial derivatives are not calculated	
NCHECK=0	program terminates without a final random check	
NRAND=0	program performs one search from specified starting point	
AREA:	area of fuel cell, cm^2/cell	1316.13
C11, C12, C13, C14:	coeffiecients used in calculating equation convergence of water shift reaction; -9.0283D+8, 3.5603D+6, 4.3662D+3, -3.0526	
CD:	current density, A/cm^2	0.1
CN1:	U*A/CMIN in heat exchanger	1.3
CK1, CK2:	coefficients used in calculating kinetic equation of demathanation reaction; 127., -4417.7	
DD1:	amount of input methane, g-mole/hour	1297.3
EXT:	exta air in burner, %	100.
NST:	number of stacks in fuel cell	6
PH_2O:	mole fraction of H_2O	0.095
PO_2:	mole fraction of O_2	0.1413
TAT:	ambient temperature, K	298.
TOPFC:	operating temperature of fuel cell, K	463.
U1:	heat transfer coefficient of heat exchangers, cal./m^2-hr-K	48825.1016
CC1, CC2:	power inverter cost data	
CC1	50., 164., 203., 248., 304., 366., 433., 528., 657., 920., 1560., 2810., 1000000000	
CC2	200., 160., 150., 140., 130., 120., 110., 100., 90., 80., 70., 60., 50	

TABLE 2 (cont'd)

INPUT DATA FOR OPTIMIZATION COMPUTER MODEL (SAMPLE RUN)

		min.	max.	init.
X(1):	utilization of H_2 in the fuel cell	0.7	0.95	0.89875
X(2):	steam/methane ratio	1.7	3.	1.7192
X(3):	number of cell plates per stack	200.	250.	225.38
S(1):	flow rate of CH_4 in steam 7; lb-mole/hr			0.51439
S(2):	CO "			1.7525
S(10):	" 9			1.3551
S(7):	" 13			0.018455
S(5):	CO_2 7			0.63024
S(17):	" 9			0.98925
S(9):	" 13			2.3257
S(8):	H_2 7			7.6278
S(12):	" 9			8.0222
S(3):	" 13			9.3582
S(11):	H_2O 7			2.5643
S(13):	" 9			2.1926
S(4):	" 13			0.8295
S(14):	temperature of stream 8; K			879.31
S(16):	" 27			637.04
S(18):	" 5			1096.6
S(15):	" 18			1197.7
S(6):	" 16			772.8
S(20):	amount of catalyst in reformer section; lb			60.

TABLE 3

ASSUMED PARAMETERS

1. input of fuel (methane) - 1297.3 g-mole/hour
2. designed AC output - 100 kW
3. average current density on the fuel cell plate - 0.1 A/cm^2
4. operating pressure in the reformer and fuel cell stack - 1 atm.
5. utilization of O_2 in the fuel cell stack - 50
6. catalyst factor (=utilization x surface area x loading) - 45
7. dimension of cell plate - 17x12 in^2
8. ADT (approach differential temperature) of shift rxn. in the reformer and shift converters - 15 K
9. utilization of air in the burner - 50
10. the bulk density of catalyst in the reformer - 30 lb/ft^3 volume of reformer
11. temperature of inlet steam - 391 K
12. operating temperature of fuel cell stack -463K

value than the minimum steam to carbon ratio (Refs. 8 and 9). There are no further restrictions for one to assign the maximum and minimum of variables, but the reasonable values will result in less computation time. The values chosen in this sample problem are listed in Table 2 under NAMELIST dataset S1. Some of the operating conditions of the powerplant model used as input to this optimization procedure are derived from the steady state performance code (Ref. 10).

The output of this sample problem is shown in Figure 7. First, the program reprints all of the input data. The optimal solutions of three design variables are given under the policy vector, and the minimal cost is the value of Y. The sample output are given below:

Hydrogen utilization (x_1)	90.6
Steam to carbon ratio (x_2)	1.7
Number of cell plates per stack (x_3)	226
Minimal equipment capital cost of 100 kW (AC) PAFC powerplant	$33534

The required CPU time to run this sample problem is less than 1 minute on IBM/370.

3.4 Discussion

It can be noted that the optimal values are some differnt from the usual design. The optimal hydrogen utilization in the stack is higher than the designed one (0.906 and 0.75-0.85, respectively), and the optimal S/C (1.7) is close to the minimum value.

```
&CON JINEQ=10,
     JEQ=0,
     N=3,
     KBOUND=2,
     KMIN=1,
     NRAND=0,
     KCLIMB=0,
     KKOFF=0,
     MPEN=0,
     IOUT=0,
     NCHECK=0,
     IGRAD=1,
     KSENS=0,
     KGRID=1,
     KSCALE=0,
     IPMAX=25,
     ALPHA=1.D+00,
     ARAND=0.314159267,
     BOUND=6.023D+23,
     CFAC=1.D-02,
     PNLTY=1.D+00,
     CONST=8.D+00,
     DELTA=1.D-10,
     DX=1.D+02,
     EFAC=1.D-03,
     EPSLN=1.D-05,
     THETA=1.D-01,
     TIME=0.,
     CONVG=1.D-4,
     GRADX=0.,
&END
&INIT DD1=1297.3,TAT=298.,PO2=0.1413,PH2O=0.095,TOPFC=463.,CN1=1.3,
U1=48825.1016,EXT=100.,NST=6,CD=0.100,AREA=1316.13,T26=391.,
&END
&CON1 CK1=127.,CK2=-4417.7,C11=-9.0283D+8,C12=3.5603D+6,C13=4.3662D+3,
C14=-3.0526,
&END
&CON2 CC1=50.,164.,203.,248.,304.,366.,433.,528.,657.,920.,1560.,
2810.,1000000000.,
CC2=200.,160.,150.,140.,130.,120.,110.,100.,90.,80.,70.,60.,
50.,
&END
&S1 XMIN(1)= 0.7,
     XMIN(2)= 1.7,
     XMIN(3)= 200.,
     XMAX(1)= 0.95,
     XMAX(2)= 3.0,
     XMAX(3)= 250.,
     X(1)=0.89875,
     X(2)=1.7192,
     X(3)=225.38,
     S(1)=0.51439,
     S(2)=1.7525,
     S(3)=9.3582,
     S(4)=0.8295,
     S(5)=0.63024,
     S(6)=772.8,
     S(7)=0.018455,
     S(8)=7.6278,
     S(9)=2.3257,
     S(10)=1.3551,
     S(11)=2.5643,
     S(12)=8.0222,
     S(13)=2.1926,
     S(14)=879.31,
     S(15)=1197.7,
     S(16)=637.04,
     S(17)=0.98925,
     S(18)=1096.6,
     S(20)=60.,
&END
```

Figure 6 Sample Input Data

```
&CON
JINEQ= 10
JEQ= 0
N= 3
KBOUND= 2
KMIN= 1
NRAND= 0
KCLIMB= 0
KKOFF= 0
MPEN= 0
IOUT= 0
NCHECK= 0
IGRAD= 1
KSENS= 0
KGRID= 1
KSCALE= 0
IPMAX= 25
ALPHA= 1.0
ARAND= 0.3141592
BOUND= 0.60230D24
CFAC= 0.10D-01
PNLTY= 1.0
CONST= 8.0
DELTA= 0.10D-09
DX= 100.0
EFAC= 0.10D-02
EPSLN= 0.10D-04
THETA= 0.10D0
TIME= 0.0
CONVG= 0.10D-03
GRADX= 0.0
&END
&INIT
DD1= 1297.30
TAT= 298.0
PO2= 0.14130
PH2O= 0.950D-01
TOPFC= 463.0
CN1= 1.30
U1= 48825.10160
EXT= 100.0
NST= 6
CD= 0.10D0
AREA= 1316.130
T26= 391.0
&END
&CON1
CK1= 127.0
CK2= -4417.70
C11= -902830000.0
C12= 3560300.0
C13= 4366.20
C14= -3.05260
&END
&CON2
CC1= 50.0, 164.0, 203.0, 248.0, 304.0, 366.0, 433.0, 528.0, 657.0, 920.0, 1560.0
2810.0, 10000000000.0
CC2= 200.0, 160.0, 150.0, 140.0, 130.0, 120.0, 110.0, 100.0, 90.0, 80.0, 70.0
60.0, 50.0
```

Figure 7 Sample Computer Run

```
$END
$S1
S= 0.514390, 1.75250, 9.35820, 0.82950, 0.630240, 772.80, 0.184550D-01, 0.762780D01
 0.232570D01, 0.135510D01, 0.256430D01, 0.802220D01, 0.219260D01, 0.879310D03
 0.119770D04, 0.637040D03, 0.989250D0, 0.109660D04, 0.0, 60.0, 5*0.0
X= 0.898750, 1.71920, 225.380, 22*0.0
XMIN= 0.70, 1.70, 200.0, 22*0.0
XMAX= 0.950, 3.0, 250.0, 22*0.0
$END

    BEST SOLUTION AFTER    1 STARTING POINTS:

    SIGMA0= 0.13102D-04

      Y= 0.33534D 05

    POLICY VECTOR:

    X(   1)= 0.90586D 00
    X(   2)= 0.17000D 01
    X(   3)= 0.22566D 03

 ********************************************************************************
```

Figure 7 (cont'd) Sample Computer Run

This program developed herein can help set the values of operation parameters in order to optimize the cost of a given plant. As an example, it might be used to perform the analysis to determine the tradeoff between operating a powerplant at a higher pressure and temperature, which improves power section efficiency, but also accelerates power section replacement schedule, and operating at lower temperature and pressure. These design and operating options are very important in establishing technology goals for the phosphoric acid fuel cell system.

3.5 Further Developments

There are two other optimization programs embedded in the overall strategy, which is not included in this manual; refer to Refs. 8 and 9. One subsystem program determines the minimum steam to carbon ratio for the reformer. This lowest permitted ratio value is a critical constraint for the total optimization model, not only as an operational danger point to the reformer catalyst, but also as cost factor to be minimized in the reaction water recovery heat exchanger subsystem. The other subsystem optimization program analyzes the thermal management heat exchanger network, and can improve that subsystem prior to the main optimization analysis of the total system.

REFERENCES

1. Alkasab, K., Presler, A., and Lu, C-Y., "Thermodynamic and Performance Model for Phosphoric Acid Fuel Cell System", Proc. 6th IASTED Int. Symp. on ENERGY '83, San Francisco, May 16-18, 1983.

2. Bauman, H.C., "Fundamentals of Cost Engineering in the Chemical Industry", Reinhold, New York, 1964.

3. Steele, R.V. et al., "Comparative Assessment of Residential Energy Supply System that Use Fuel Cells", Rept. of SRI International, Menlo Park, CA, EPA-600/7-79-105b, April 1979.

4. Hooke, R. and Jeeves, "Direct Search Solution of Numerical and Statistical Problem", J. Assoc. Comp., Mach., 8, No. 2, 1961.

5. Lu, C-Y, "Global Optimization in Engineering Design Problem", Master Thesis, University of Cincinnati, June 1980.

6. Gottfried, B.S., and Weisman, J., "Introduction to Optimization Theory", Prentice Hall, New Jersey.

7. Keefer, D.L., and Gottfried, B.S., "Differential Constraint Scaling and Statistical Problems", J. Assoc. Comp., Mach., 8, No. 1, 1962.

8. Lu, C-Y., "Transient Responses of Phosphoric Acid Fuel Cell Powerplant System", Ph.D. Disertation, Cleveland State University, December 1983.

9. Alkasab, K.A., Lu, C-Y., and Presler, A.F., "Optimized Cost and Performance Model of a Phosphoric Acid Fuel Cell Powerplant", Proc. of 8th IASTED International Symposium on Energy '83, Orlando, November 9-11, 1983.

10. Alkasab, K.A., and Lu, C-Y., "Phosphoric Acid Fuel Cell Powerplant Performance Model and Computer Program", NASA CR-174638, January 1984.

LISTING OF THE OPTIMIZATION COMPUTER MODEL

```
      BLOCK DATA
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GS(7),HS(7),HCAS(7),HCBS(7),HCCS(7)
      COMMON /ETHDA/ GS,HS,HCAS,HCBS,HCCS
C STANDARD VALUE OF FREE ENERGY, ENTHALPHY AND HEAT CAPACITY CONSTANT
      DATA GS/-12140.,0.,-32781.,-94258.,0.,-54635.,0./,                -
     1HS/-17889.,0.,-26416.,-94051.,0.,-57798.,0./,                     -
     2HCAS/3.381,6.148,6.42,6.214,6.947,7.256,6.524/,                   -
     3HCBS/0.018044,0.003102,0.001665,0.010396,-0.0002,0.002298,0.00125 -
     4/,HCCS/-0.0000043,-0.000000923,-.000000196,-.000003545,0.000000481-
     50,0.000000283,-0.000000001/
      END
C
C*****************************************************************
C GIVEN PARAMETERS
C*****************************************************************
C1).INPUT OF METHANE; 1297.3 G-MOLE/HR
C2).KW A.C. FOR DESIGNING; 100 KW
C3).AVERAGE CURRENT DENSITY IN FUEL CELL; 0.100 A/CM**2
C4).OPERATING PRESSURE IN REFORMER AND FUEL CELL; 1 ATM
C5).UTILIZATION OF O2 IN FUEL CELL; 50%
C6).CATALYST FACTOR(=CATALYST UTILIZATION*CATALYST SURFACE AREA*CATALYST
CING); 45
C7).DIMENSION OF CELL PLATE; 17*12 IN**2
C8).ADT(APPROACH DIFFERENTIAL TEMPERATURE) OF WATER SHIFT IN THE REFORME
CSHIFT CONVERTERS; 15 C
C9).UTILIZATION OF AIR IN THE BURNER; 50%
C10).THE BULK DENSITY OF CATALYST IN THE REFORMER; 30 LB/FT**3 VOLUME OF
CMER SECTION
C11).TEMP. OF INLET STEAM; 391 K
C12).OPERATING TEMP. OF FUEL CELL IS 463 K
C
C
C*****************************************************************
C VARIABLES
C*****************************************************************
C          DESCRIPTION                              MIN.          MAX.
C1).UTILIZATION OF H2 IN THE FUEL CELL **********  0.7     X(1)  0.95
C2).STEAM/METHANE RATIO ********************       1.7     X(2)    3.
C3).NO. OF CELL PLATES PER STACK ***************  200.     X(4)  250.
C
C*****************************************************************
CTRIAL AND ERROR VARIABLES
C*****************************************************************
C
C1).FLOW RATE OF CH4 IN STREAM 7; LB-MOLE/HR                  S(1)
C2).FLOW RATE OF CO IN STREAM 7;  LB-MOLE/HR                  S(2)
C3).          "               9     "                         S(10)
C4).          "               13    "                         S(7)
C5).FLOW RATE OF CO2 IN STREAM 7; LB-MOLE/HR                  S(5)
C6).          "               9     "                         S(17)
C7).          "               13    "                         S(9)
C8).FLOW RATE OF H2 IN STREAM 7; LB-MOLE/HR                   S(8)
```

```
0006500 C9).          "                9      "                          S(12)
0006600 C10).         "               13      "                          S(3)
0006700 C11).FLOW RATE OF H2O IN STREAM 7; LB-MOLE/HR                    S(11)
0006800 C12).         "                 9     "                          S(13)
0006900 C13).         "                 13    "                          S(4)
0007000 C14).TEMP. OF STREAM 8; K                                        S(14)
0007100 C15).TEMP  OF STREAM 27; K                                       S(16)
0007200 C16).         "       5; K                                       S(18)
0007300 C17).         "       18; K                                      S(15)
0007400 C18).         "      16; K                                       S(6)
0007500 C19).CATALYST OF REFORMER SECTION; LB                            S(20)
0007600 C
0007700 C
0007800 C
0007900 C*************************************************************************
0008000 C OBJECTIVE FUNCTION
0008100 C*************************************************************************
0008200 C1).COST OF REFORMER,F(AMONT OF CATALYST)
0008300 C2).COST OF SHIFT CONVERTERS(HIGH TEMP. AND LOW TEMP.),F(FLOW RATE OF H2
0008400 C3).COST OF FUEL CELL,F(NO. OF CELL PLATES)
0008500 C4).COST OF.HEAT EXCHANGERS,F(TRANSFER AREA)
0008700 C5).COST OF POWER INVERTER
0008800 C
0008900 C*************************************************************************
0009000 C CONSTRAINTS
0009100 C*************************************************************************
0009200 C1).MOLE COMPOSITION OF CO IN THE INLET OF FUEL CELL     <1%
0009300 C2).0.5 VOLT < VOLTAGE OF EACH CELL <0.75 VOLT
0009400 C3).OUTPUT VOLTAGE OF EACH STACK       <300 V D.C.
0009500 C4).A.C. OUTPUT > 100 KW
0009600 C5).125 C < FUEL INLET TEMP. < 204 C
0009700 C6).90  C < OXIDENT INLET TEMP. < 149 C
0009800 C7). MOLE COMPOSITION OF H2 IN THE INLET STREAM OF FUEL CELL >60%
0009900 C
0010000 C
0011700 C*************************************************************************
0011800 C
0011900 C CONSTRAINED OPTIMIZATION USING MIXED PENALTIES AND UNCONSTRAINED
0012000 C     TECHNIQUES FOR EXTREMIZATION ("COMPUTE")
0012100 C
0012200 C*************************************************************************
0012300 C
0012400       IMPLICIT REAL*8(A-H,O-Z)
0012500       REAL ARAND,XR
0012600       DIMENSION G(30),GB(30),PD(30),SC(30),X(25),X0(25),X1(25),XMIN(25),-
0012700      1          XMAX(25),XSTAR(25),XR(25),S(25),XST12(25),S(25)          -
0012800      2,CC1(13),CC2(13)
0012900       NAMELIST/CON/ JINEQ,JEQ,N,KBOUND,KMIN,NRAND,KCLIMB,KKOFF,MPEN,    -
0013000      1IOUT,NCHECK,IGRAD,KSENS,KGRID,KSCALE,IPMAX,ALPHA,ARAND,BOUND,CFAC -
0013100      2,PNLTY,CONST,DELTA,DX,EFAC,EPSLN,THETA,TIME,CONVG,GRADX
0013200       NAMELIST/S1/ S,X,XMIN,XMAX
0013300       NAMELIST/INIT/DD1,TAT,PO2,PH2O,TOPFC,CN1,U1,EXT,NST,CD,AREA,T26
0013400       NAMELIST/CON1/CK1,CK2,C11,C12,C13,C14
0013500       NAMELIST/CON2/CC1,CC2
```

```
      COMMON DELTA,DX,EFAC,EPSLN,G,GRADX,PNLTY,SC,SIGMA0,SIGMA1,SIGMA2, -
     1       TIME,X,XMIN,XMAX,Y,Z
      COMMON IGRAD,IOBJ,IOUT,IT,ITMAX,JEQ,JINEQ,KBOUND,KCLIMB,KMIN,M,N, -
     1       NTEST
      COMMON/DLK/PD,GB,SIGLPL,ALPHA,CFAC,KSCALE,IP,MPEN
      COMMON/ID/ID,ILUY
      COMMON/NOTI/NOTI
      COMMON/S2/ S
      COMMON/AG/ BAT,ALPH
      COMMON/CONN/CK1,CK2,C11,C12,C13,C14
      COMMON/INTL/DD1,TAT,PO2,PH2O,TOPFC,EXT,CD,AREA,T26,NST
      COMMON/CN/CN1/U/U1
      COMMON/CONM/CC1,CC2
C*************************************************************************
C DEFINITION:
C N: NUMBER OF INDEPENDENT VARIABLES
C JEQ: NUMBER OF FUNCTIONAL EQUALITY CONSTRAINTS
C JINEQ: NUMBER OF FUNCTIONAL INEQUALITY CONSTRAINTS
C M: NUMBER OF FUNCTIONAL CONSTRAINTS
C IPMAX: MAX. NUMBER OF PENALTY LEVELS
C IOBJ: FUNCTION EVALUATION COUNTER
C IP: PENALTY LEVEL COUNTER
C IR: RANDOM CHECK IMPROVEMENT COUNTER
C IT: SEARCH ITERATION COUNTER
C ITMAX: MAX. NUMBER OF ITERATIONS PER GRADIENT SEARCH
C ITOBJ: TOTAL NUMBER OF FUNCTIONAL EVALUATIONS
C ITOT: TOTAL NUMBER OF SEARCH ITERATIONS
C NTEST: NUMBER OF RANDOMLY GENERATED EXPLORATORY POINTS IN THE PATTERN
C ALPHA: SCALING FACTOR BETWEEN SIGMA2 AND SIGMA1
C ARAND: INITIAL PARAMETER FOR THE RANDOM NUMBER GENERATOR
C CFAC: PROPORTIONALITY CONSTANT USED IN CAL. PNLTY
C CONST: PENALTY MULTIPLIER
C CONVG: PENALTY LEVEL CONVERGENCE CRITERION
C DELTA: FINITE-DIFFERENCE INCREMENT
C DX: LENGTH OF A GRADIENT MOVE
C EFAC: SEARCH INCREMENT REDUCTION FACTOR
C EPSLN: GRADIENRT VECTOR CONVERGENCE FACTOR
C GRADX: GRADIENT VECTOR CONVERGENCE CRITERION
C THETA: MAX. TIME ALLOWED TO EVALUATE THE OBJECTIVE FUNCTION WHEN CAL.
C        GRADIENT VECTOR BY CENTRAL DIFFERENCES
C GB(I): UNSCALED VALUE OF CONSTRAINT I
C G(I): SCALED VALUE OF CONSTRAINT I
C SC(I): SCALING FACTOR FOR CONSTRAINT I AUTOMATICALLY DETERMINED AT THE
C        NING OF EACH PENALTY LEVEL: G(I)=GB(I)*SC(I)
C PD(I): SUM OF THE MAGNITUDES OF D(GB(I))/(D(X(KFR)),KFR=1,2,...,N
C SIGMA0: SUM OF THE SQUARES OF THE CONSTRAINTS WHICH ARE VIOLATED (EXCL
C         BOUNDS ON THE INDEPENDENT VARIABLES)
C SIGMA1: SUM OF THE SQUARES OF THE SCALED CONSTRAINTS WHICH ARE VIOLATE
C         EXCLUDING THE INDEPENDENT VARIABLE BOUNDS)
C SIGMA2: PENALTY FUNCTION COMPOSED OF THE UPPER AND LOWER BOUNDS ON THE
C         ENDENT VARIABLES
C SIGMA: SUM OF SIGMA0 AND A MODIFIED SIGMA2 OBTAINED BY DIVIDING SIGMA2
C        THE PENALTY COEFF. SQUARED
C TMTOT: TOTAL SEARCH TIME
```

```
0019000 C
0019100 C PROGRAM OPTIONS:
0019200 C IOUT=0 OUTPUT FOLLOWS FINAL OPTIMIZATION
0019300 C IOUT=1 OUTPUT FOLLOWS OPTIMIZATION FOR EACH STARTING POINTY
0019400 C IOUT=2 OUTPUT FOLLOWS OPTIMIZATION FOR EACH PENALTY LEVEL
0019500 C IOUT=3 OUTPUT FOLLOWS EACH MAJOR SEARCH OR GRADIENT MOVE
0019600 C KBOUND=0 INDEPENDENT VARIABLES ARE UNBOUNDED
0019700 C KBOUND=1 INDEPENDENT VARIABLES ARE NONNEGATIVE
0019800 C KBOUND=2 INDEPENDENT VARIABLES ARE BOUNDED BY SPECIFIC VALUES
0019900 C KCLIMB=0 MODIFIED OBJECTIVE FUNCTION OPTIMIZED VIA HOOKE AND JEEVES PA
0020000 C          SEARCH
0020100 C KGRID=0 PROGRAM USES FINE-GRID HILLCLIMBING CONVERGENCE CRITERION AT A
0020200 C          PENALTY LEVELS
0020300 C KGRID>0 PROGRAM USES FINE-GRID HILLCLIMBING CONVERGENCE CRITERION AT E
0020400 C          PENALTY LEVELS
0020500 C KKOFF=0 AUTOMATIC TERMINATION IF Z NOT UNIQUELY DETERMINED BY X-VECTOR
0020600 C KKOFF>0 COMPUTATION CONTINUES IF Z NOT UNIQUELY DETERMINED BY X-VECTOR
0020700 C KMIN=0 PROGRAM MAX. THE OBJECTIVE FUNCTION
0020800 C KMIN=1 PROGRAM MIN. THE OBJECTIVE FUNCTION
0020900 C KSCALE=0 CONSTRAINTS SCALED AUTOMATICALLY
0021000 C KSCALE=1 CONSTRAINTS NOT SCALED
0021100 C KSCALE=2 CONSTRAINTS NOT SCALED DURING A PENALTY LEVEL DUE TO DETECTIO
0021200 C          OF SUSPECTED CONSTRAINT RIDGE BY AUTOMATIC TESTING PROCEDURE
0021300 C KSENS=0 FINAL VALUES OF PARTIAL DERIVATIVES ARE NOT CAL.
0021400 C KSENS>0 PARTIAL DERIVATIVES OF OBJECTIVE FUNCTION ARE CAL. AFTER OPTIM
0021500 C         HAS BEEN COMPLETED
0021600 C NCHECK=0 PROGRAM TERMINATES WITHOUT A FINAL RANDOM CHECK
0021700 C NCHECK>0 PROGRAM GENERATES 'NCHECK' THOUSAND RANDOM CHECK POINTS AFTER
0021800 C          THE OPTIMUM
0021900 C NRAND=0 PROGRAM PERFORMS ONE SEARCH FROM SPECIFIED STARTING POINT
0022000 C NRAND>0 PROGRAM PERFORMS 'NRAND' SEARCHES FROM SPECIFIED STARTING POIN
0022100 C
0022200 C DEFINITION OF VARIABLES USED IN 'OBJECT':
0022300 C AREA: AREA OF FUEL CELL, CM**2/CELL
0022400 C C11,C12,C13,C14: COEFF. USED IN CALCULATING EQU. CON. OF WATER SHIFT R
0022500 C CD: CURRENMT DENSITY, AMP/CM**2
0022600 C CN1: Q*A/CMIN IN HEAT EXCHANGER
0022700 C CK1,CK2: COEFF. USED IN CALCULATING KINETIC EQUATION OF DEMETHANATION
0022800 C DD1: AMOUNT OF INPUT METHANE, G-MOLE/HR
0022900 C EXT: EXTRA AIR IN BURNER
0023000 C NST: NUMBER OF STACKS IN FUEL CELL
0023100 C PH2O: MOLE FRACTION OF H2O
0023200 C PO2: MOLE FRACTION OF O2
0023300 C TAT: AMBIENT TEMP., K
0023400 C TOPFC: OPERATING TEMP. OF FUEL CELL, K
0023500 C U1:TOTAL HEAT TRANSFER COEFF. OF HEAT EXCHANGERS(ASSUME THE SAME U VAL
0023600 C    ALL EXCHANGERS AND CONDENSER),  CAL./M**2-HR-C
0023700 C*************************************************************************
0023800 C
0023900  2001 FORMAT('0',4X,96('*')/)
0024000  2015 FORMAT('     X(',I4,')=',D12.5)
0024100  3003 FORMAT('0    BEST SOLUTION AFTER ',I4,' STARTING POINTS:'/,'0    S-
0024200     1IGMA0=',D12.5/,'0     Y=',D12.5/,'0    POLICY VECTOR:'/)
0024300  3050 FORMAT('     PENALTY LEVEL CONVERGENCE CRITERION CANNOT BE SATISFI-
```

```
0024400      1ED'/)
0024500  3051 FORMAT('0    APPARENT CONSTRAINT RIDGE ENCOUNTERED - DISCONTINUE S-
0024600      1CALING'/,'0    SCALING WILL BE RESUMED IF THE NONSCALING PROCEDURE-
0024700      2 FAILS TO DECREASE SIGMA0'/)
0024800  3055 FORMAT('0    NONE OF THE',I4,' SOLUTIONS SAATISFY THE PENALTY LEVE-
0024900      1 CONVERGENCE CRITERION'/)
0025000  3056 FORMAT('1    PROBLEM VIOLATES BASIC UNDERLYING ASSUMPTION THAT OBJ-
0025100      1ECTIVE FUNCTION IS UNIQUELY'/,'     DEFINED BY VECTOR X -- VALUES -
0025200      2ON SUCCESSIVE EVALUATIONS:'/,                                    -
0025300      3'     Z1=',D23.16,'     ZSAVE=',D23.16/,                         -
0025400      4'     Y1=',D23.16,'     YSAVE=',D23.16//)
0025500 C ESTABLISH NUMERICAL VALUE FOR PROGRAM CONSTANTS
0025600       READ(5,CON)
0025700       WRITE(6,CON)
0025800 C READ THE INITIAL INPUT DATA FOR CAL. FUEL CELL SYSTEM IN 'OBJECT'
0025900       READ(5,INIT)
0026000       WRITE(6,INIT)
0026100 C READ THE CONST. USED IN FUEL CELL SYSTEM IN 'OBJECT'
0026200       READ(5,CON1)
0026300       WRITE(6,CON1)
0026400 C READ THE COST DATA OF POWER INVERTER
0026500       READ(5,CON2)
0026600       WRITE(6,CON2)
0026700       ILUY=0
0026800       DO 80 I=1,25
0026900    80 S(I)=0.
0027000 C READ INITIAL VALUE OF VARIABLES
0027100       READ(5,S1)
0027200       WRITE(6,S1)
0027300       IF(KGRID .EQ. 1) EFAC=1.D-03
0027400       NCHECK=NCHECK/1000
0027500       IF (N.EQ.0) STOP
0027600       IF(KSCALE.EQ.2) KSCALE=0
0027700       XN=N
0027800       IF(GRADX.EQ.0.) GRADX=XN*EPSLN
0027900       GRADX0=GRADX
0028000       NTEST=10*N
0028100       ITMAX=10*N
0028200       IF(ITMAX .GT. 250) ITMAX=250
0028300       IF(ITMAX .LT. 50) ITMAX=50
0028400       M=JEQ+JINEQ
0028500       IF(M .LE. 1) KSCALE=1
0028600       XM=M
0028700       M1=M+1
0028800       N1=N+1
0028900       NCHECK=1000*NCHECK
0029000       IF(CONVG .GT. 0.) GO TO 8005
0029100       CONVG=XM*EPSLN
0029200       IF(KCLIMB .EQ. 0) GO TO 8005
0029300       IF(KBOUND .EQ. 1) CONVG=CONVG+XN*EPSLN
0029400       IF(KBOUND .EQ. 2) CONVG=CONVG+XN*EPSLN*2.D+00
0029500  8005 CONTINUE
0029600 C READ OPTIONAL INPUT IN SUBROUTINE OBJECT
0029700       CALL OBJECT(X,Y1,Z1,1)
```

```
0029800        SHIFT=0.
0029900        IF(KCLIMB .GT. 0) SHIFT=DELTA
0030000        IF(KBOUND-1) 2,4,9
0030100      2 DO 3 I=1,N
0030200      3 XMIN(I)=-BOUND
0030300        GO TO 6
0030400      4 DO 5 I=1,N
0030500      5 XMIN(I)=1.D-05
0030600      6 DO 7 I=1,N
0030700      7 XMAX(I)=BOUND
0030800      9 IF(NRAND .GT. 0) GO TO 11
0030900        DO 10 I=1,N
0031000        XITEMP=DMAX1(X(I),(XMIN(I)+SHIFT))
0031100        X(I)=DMIN1(XITEMP,(XMAX(I)-SHIFT))
0031200     10 CONTINUE
0031300     11 IF (KBOUND.LT.2) GO TO 100
0031400        SUM=0.
0031500        DO 12 I=1,N
0031600        IF(XMIN(I).LE.XMAX(I)) GO TO 12
0031700        GO TO 9999
0031800     12 SUM=SUM+(XMAX(I)-XMIN(I))**2
0031900        DX=DSQRT(SUM)
0032000    100 CONTINUE
0032100        IRAND=1
0032200        YMAX=-1.D+75
0032300 C INITIALIZE COUNTERS AND PARAMETERS
0032400    300 TMTOT=0.
0032500        ITOBJ=0
0032600        ITOT=0
0032700        GRADX=GRADX0
0032800        IP=1
0032900        IF(KSCALE.EQ.2) KSCALE=0
0033000        XSIG0=0.
0033100        XSIG2=0.
0033200 C TEST TO DETERMINE WHETHER OBJECTIVE FUNCTION IS UNIQUELY DEPENDENT UPO
0033300    520 CALL OBJECT(X,Y1,Z1,2)
0033400        YSAVE=Y1
0033500        ZSAVE=Z1
0033600        CALL OBJECT(X,Y1,Z1,2)
0033700    215 ERMAIN=DABS(Z1-ZSAVE)
0033800        IF(ERMAIN .LT. DELTA) GO TO 220
0033900        IF(KMIN .EQ. 0) GO TO 216
0034000        Z1=-Z1
0034100        ZSAVE=-ZSAVE
0034200        Y1=-Y1
0034300        YSAVE=-YSAVE
0034400        WRITE(6,3056) Z1,ZSAVE,Y1,YSAVE
0034500    216 CONTINUE
0034600        IF(KKOFF .EQ. 0) CALL OBJECT(X,Y1,Z1,3)
0034700        IF(KKOFF .EQ. 0) STOP
0034800        IF(KMIN .EQ. 0) GO TO 220
0034900        Z1=-Z1
0035000        ZSAVE=-ZSAVE
0035100        Y1=-Y1
```

```
0035200       YSAVE=-YSAVE
0035300   220 CONTINUE
0035400       CALL OBJECT(X,Y1,Z1,4)
0035500       Y=Y1
0035600       Z=Z1
0035700       IF((M.EQ.0).AND.(KBOUND*KCLIMB .EQ. 0)) GO TO 401
0035800 C DETERMINE INITIAL VALUE OF PENALTY COEFFICIENT
0035900   701 CALL COEFF(0)
0036000       IF(PNLTY.LT.1.) PNLTY=1.
0036100       CALL OBJECT(X,Y1,Z1,2)
0036200       SIGMA=SIGMA0+SIGMA2/(ALPHA*PNLTY**2)
0036300       Y=Y1
0036400       Z=Z1
0036500   302 IF(KGRID .EQ. 0) GO TO 401
0036600 C DETERMINE COARSE-GRID CONVERGENCE CRITERION
0036700       EFAC=1.
0036800       IF(PNLTY .LT. 1.D-06*SIGMA1) GO TO 401
0036900       EFAC=1.D-01
0037000       IF(PNLTY .LT. SIGMA1) GO TO 401
0037100       EFAC=0.01
0037200       IF(PNLTY .GE. 1.D06*SIGMA1) EFAC=1.D-03
0037300 C PERFORM SEARCH
0037400   401 CALL PATTRN
0037500   404 ITOBJ=ITOBJ+IOBJ
0037600       ITOT=ITOT+IT
0037700       TMTOT=TMTOT+TIME
0037800       SIGMA=SIGMA0+SIGMA2/(ALPHA*PNLTY**2)
0037900       IF(KMIN .EQ. 0) GO TO 405
0038000       Y=-Y
0038100       Z=-Z
0038200   405 IF((M .EQ. 0).AND.(KBOUND*KCLIMB .EQ. 0)) GO TO 600
0038300 C TEST FOR CONVERGENCE
0038400   409 CONTINUE
0038500       IF(SIGMA .GT. CONVG) GO TO 415
0038600       IF(KGRID .EQ. 0) GO TO 600
0038700       IF(EFAC .LE. 1.D-03) GO TO 600
0038800       EFAC=1.D-03
0038900       IF(KMIN .EQ. 0) GO TO 410
0039000       Y=-Y
0039100       Z=-Z
0039200   410 GO TO 401
0039300   415 IF (M .NE. 0) GO TO 420
0039400       IF(DABS(SIGMA2-XSIG2)-DELTA) 590,590,425
0039500   420 YLIM=DABS(Y*1.D06)
0039600       IF(YLIM .LT. 1.D06) YLIM=1.D06
0039700       IF ((PNLTY*SIGMA1+SIGMA2/(ALPHA*PNLTY**2) .GT. YLIM) .AND.      -
0039800      1(DABS(SIGMA0-XSIG0) .LT. DELTA).AND.(DABS(SIGMA2-XSIG2).LE.DELTA))-
0039900      2 GO TO 590
0040000       IF(IP.GE.IPMAX) GO TO 800
0040100   425 CONTINUE
0040200 C TEST FOR "FALSE STOP" ON CONSTRAINT RIDGE
0040300 C IF CONSTRAINT-RIDGE PROBLEM ENCOUNTERED USING AUTOMATIC SCALING, SCALI
0040400 C NOT USED IN NEXT PENALTY LEVEL AND NOT USED IN SUBSEQUENT PENALTY LEVE
0040500 C UNLESS LACK OF SCALING FAILS TO REDUCE SIGMA0
```

```
0040600 C SCALING MAY AUTOMATICALLY BE SWITCHED OFF AND ON SUBSEQUENTLY
0040700       IF(KSCALE-1) 427,430,426
0040800   426 IF(SIGMA0 .LT. XSIG0) GO TO 430
0040900       KSCALE=0
0041000       GO TO 430
0041100   427 IF( SIGMA0 .LE. CONVG) GO TO 430
0041200       IF(PNLTY*SIGMA1 .LT. 1.0D01*DABS(Y)) GO TO 430
0041300       IF((DABS(SIGMA0-XSIG0) .GT. DELTA).OR.(DABS(SIGMA2-XSIG2) .GT.  -
0041400      1DELTA)) GO TO 430
0041500       IOUT=3
0041600       KSCALE=2
0041700       IF(IOUT .LT. 2) GO TO 430
0041800       WRITE(6,3051)
0041900       WRITE(6,2001)
0042000       IOUT=0
0042100   430 CONTINUE
0042200       XSIG0=SIGMA0
0042300       XSIG2=SIGMA2
0042400 C INCREASE PENALTY COEFFICIENT AND GRADIENT CONVERGENCE CRITERION
0042500       IF(PNLTY*SIGMA1 .GT. DABS(Y-SIGMA2/(ALPHA*PNLTY**2))) GRADX=CONST*-
0042600      1GRADX
0042700       PNLTY=CONST*PNLTY
0042800       IP=IP+1
0042900 C RESCALE CONSTRAINTS
0043000       CALL OBJECT(X,Y1,Z1,4)
0043100       Y=Y1
0043200       Z=Z1
0043300       GO TO 302
0043400   590 WRITE(6,3050)
0043500       NOTI=1
0043600 C PROBLEM TERMINATION-OPTIMAL FEASIBLE SOLUTION HAS BEEN FOUND
0043700   600 IF(IOUT .LT. 1) GO TO 605
0043800   605 IF(KMIN .EQ. 1) Y=-Y
0043900  2013 FORMAT(1X,D12.5)
0044000       IF(Y .LE. YMAX) GO TO 607
0044100       IF((NRAND .GT. 1).AND.(SIGMA .GT. CONVG)) GO TO 607
0044200       YMAX=Y
0044300       SIGMIN=SIGMA0
0044400   607 IF (KMIN .EQ. 1) Y=-Y
0044500       IF(IRAND .GE. NRAND) GO TO 800
0044600 C WRITE FINAL RESULT
0044700   800 IF(YMAX .NE. -1.D+75) GO TO 801
0044800       WRITE(6,3055) IRAND
0044900       GO TO 802
0045000   801 IF(KMIN .EQ. 1) YMAX=-YMAX
0045100       WRITE(6,3003) IRAND,SIGMIN,YMAX
0045200       WRITE(6,2015) ((I,    X(I)),I=1,N)
0045300   802 WRITE(6,2001)
0045400   566 CONTINUE
0045500  9999 STOP
0045600       END
0045700       SUBROUTINE OBJECT(X,Y,Z,K)
0045800 C*****************************************************************
0045900 C THIS SUBROUTINE COMPUTES THE MODIFIED OBJECTIVE FUNCTION
```

```
0046000 C
0046100 C Y IS THE ACTUAL OBJECTIVE FUNCTION
0046200 C Z IS THE MODIFIED OBJECTIVE FUNCTION
0046300 C
0046400 C*************************************************************************
0046500       IMPLICIT REAL*8(A-H,O-Z)
0046600       REAL K1,K2
0046700       DIMENSION G(30),GB(30),PD(30),SC(30),X(25),XMIN(25),XDUMMY(25),   -
0046800      1XMAX(25),COEE(4),S(25)
0046900       DIMENSION CC1(13),CC2(13),GS(7),HS(7),HCAS(7),HCBS(7),HCCS(7)
0047000 C*************************************************************************
0047100 C INSERT OPTIONAL DIMENSION STATEMENTS
0047200 C*************************************************************************
0047300       DIMENSION HA(10),DNSC(7),DNSH(7),DNS1(7),DNS2(7),H(18)
0047400       COMMON DELTA,DX,EFAC,EPSLN,G,GRADX,PNLTY,SC,SIGMA0,SIGMA1,SIGMA2, -
0047500      1       TIME,XDUMMY,XMIN,XMAX,YDUMMY,ZDUMMY
0047600       COMMON IGRAD,IOBJ,IOUT,IT,ITMAX,JEQ,JINEQ,KBOUND,KCLIMB,KMIN,M,N, -
0047700      1       NTEST
0047800       COMMON/DLK/PD,GB,SIGLPL,ALPHA,CFAC,KSCALE,IP,MPEN
0047900       COMMON/CONN/CK1,CK2,C11,C12,C13,C14
0048000       COMMON/INTL/DD1,TAT,P02,PH20,TOPFC,EXT,CD,AREA,T26,NST
0048100       COMMON/CONM/CC1,CC2
0048200       COMMON/CN/CN/U/U
0048300       COMMON/ID/ID,ILUY
0048400       COMMON/AG/ BAT,ALPH
0048500       COMMON/S2/S
0048600       COMMON/ETHDA/GS,HS,HCAS,HCBS,HCCS
0048700       GO TO (9000,9900,9900,9900,9999,9900),K
0048800  9000 CONTINUE
0048900       DO 9005 I=1,M
0049000  9005 SC(I)=1.D00
0049100       IF(M .EQ. 0) SIGMA0=0.
0049200       RETURN
0049300  9900 CONTINUE
0049400 C INITIALIZE FLAG FOR SCALING--ISCALE
0049500       ISCALE=1
0049600       SIGLPL=SIGMA1
0049700       IF(IP .EQ. 1) SIGLPL=0.0
0049800  8000 CONTINUE
0049900       DO 80 KFR=1,N
0050000       IF(ISCALE .NE. 3) GO TO 8002
0050100 C INCREMENT THE INDEPENDENT VARIABLES FOR COMPUTATION OF PARTIAL DERIVAT
0050200       XXSAVE=X(KFR)
0050300       IF (DABS(X(KFR)) .LE. 1.0D00) GO TO 8001
0050400       X(KFR)=X(KFR)*(1.0D00+DELTA)
0050500       CDELTA=DELTA*X(KFR)
0050600       GO TO 8002
0050700  8001 X(KFR)=X(KFR)+DELTA
0050800       CDELTA=DELTA
0050900  8002 CONTINUE
0051000 C*************************************************************************
0051100 C INSERT STATEMENTS DEFINING THE CONSTRAINTS
0051200 C
0051300 C THE EQUALITY CONSTRAINTS G(1),G(2),..., G(JEQ) MUST PRECEDE THE
```

```
0051400 C     INEQUALITY CONSTRAINTS G(JEQ+1),G(JEQ+2),..., G(M)
0051500 C
0051600 C THE INDEPENDENT VARIABLES MUST BE CALLED X(1),X(2),..., X(N)
0051700 C*************************************************************
0051800 C      SAMPLE PROBLEM
0051900       ILUY=ILUY+1
0052000       NG=0
0052100   902 NF=0
0052200   802 NC=0
0052300   56  NE=0
0052400   402 ND=0
0052500   602 DO 1 I=2,7
0052600   1   DNSC(I)=0.
0052700       DNSC(1)=DD1
0052800       DNSH(1)=453.6*S(1)
0052900       DNSH(2)=0.
0053000       DNSH(3)=453.6*S(10)
0053100       DNSH(4)=453.6*S(17)
0053200       DNSH(5)=453.6*S(12)
0053300       DNSH(6)=453.6*S(13)
0053400       DNSH(7)=0.
0053500       CALL HEXC1(S(14),DNSH,DNSC,TAT,T10,T2,HA(1))
0053600       DO 2 I=1,7
0053700   2   DNS1(I)=DNSC(I)
0053800       DO 3 I=1,7
0053900   3   DNS2(I)=0.
0054000       DNS2(6)=DD1*X(2)
0054100       CALL DMIX1(DNS1,DNS2,T2,S(16),T3)
0054200       DNSC(6)=DNS2(6)
0054300       DNSH(1)=453.6*S(1)
0054400       DNSH(3)=453.6*S(2)
0054500       DNSH(4)=453.6*S(5)
0054600       DNSH(5)=453.6*S(8)
0054700       DNSH(6)=453.6*S(11)
0054800       CALL HEXC1(S(18),DNSH,DNSC,T3,T8,T4,HA(2))
0054900       H(1)=S(14)-T8
0055000       IF(DABS(H(1)).LT.0.2) GO TO 550
0055100       IF(ND.NE.0) GO TO 601
0055200       X14=S(14)
0055300       TT8=T8
0055400       S(14)=T8
0055500       ND=1
0055600       GO TO 602
0055700   601 XX14=(X14*T8- TT8 *S(14))/(X14-S(14)+T8-TT8)
0055800       X14=S(14)
0055900       TT8=T8
0056000       S(14)=XX14
0056100       GO TO 602
0056200   550 DNSH(1)=0.
0056300       DNSH(2)=453.6*(2.*S(1)+0.5*S(7)+0.5*S(3)*(1.0-X(1)))
0056400       DNSH(3)=0.
0056500       DNSH(4)=453.6*(S(1)+S(7)+S(9))
0056600       DNSH(5)=0.
0056700       DNSH(6)=453.6*(2.*S(1)+0.5*S(3)*(1.0-X(1)))
```

```
0056800       DNSH(7)=3411.07*(0.5*S(7)+0.5*S(3)*(1.0-X(1)) +2.*S(1))
0056900       CALL HEXC1(S(15),DNSH,DNSC,T4,T19,T5,HA(3))
0057000       H(2)=T5-S(18)
0057100       IF(DABS(H(2)).LT.0.2) GO TO 551
0057200       IF(NE.NE.0) GO TO 701
0057300       X18=S(18)
0057400       TT5=T5
0057500       S(18)=T5
0057600       NE=1
0057700       GO TO 402
0057800   701 XX18=(X18*T5- TT5 *S(18))/(X18-S(18)+T5-TT5)
0057900       X18=S(18)
0058000       TT5=T5
0058100       S(18)=XX18
0058200       GO TO 402
0058300   551 F5=0.
0058400       T=T5
0058500       K1=CK1*DEXP(CK2/T)
0058600       DO 11 I=1,7
0058700   11  F5=F5+DNSC(I)/453.6
0058800       X1=0.08154*(-(F5+   K1*S(20)) +DSQRT((F5+   K1*S(20)) **2+22.915  -
0058900      1*K1*S(20)))
0059000       T=(T5-15.)*1.8
0059100       K2=DEXP(C11/T**3+C12/T**2+C13/T+C14)
0059200       A=(K2-1.)*X1**2
0059300       B=-(3.*X1**2+K2*X(2)*X1)
0059400       C=-K2*X1**2+K2*X(2)*X1
0059500       X2=(-B-DSQRT(B**2-4.*A*C))/2./A
0059600       FCH=DD1*(1.-X1)/453.6
0059700       FCO=DD1*(X1-X1*X2)/453.6
0059800       FCO2=DD1 *X1*X2/453.6
0059900       FH2O=DD1*(X(2) -X1-X1*X2)/453.6
0060000       FH2=DD1*(3.*X1+X1*X2)/453.6
0060100       H(3)=S(1)-FCH
0060200       H(4)=FCO-S(2)
0060300       H(5)=FCO2-S(5)
0060400       H(6)=FH2O-S(11)
0060500       H(7)=FH2-S(8)
0060600       DO 552 IOP=3,7
0060700       IF(DABS(H(IOP)).GT.0.02) GO TO 553
0060800   552 CONTINUE
0060900       GO TO 554
0061000   553 S(1)=(FCH+S(1))/2.
0061100       S(2)=(FCO+S(2))/2.
0061200       S(5)=(FCO2+S(5))/2.
0061300       S(8)=(FH2+S(8))/2.
0061400       S(11)=(FH2O+S(11))/2.
0061500       GO TO 56
0061600   554 T5=T5*1.8-460.
0061700       T16=S(6)*1.8-460.
0061800       FO2=2.*(2.*FCH+0.5*S(7)+0.5*S(3)*(1.0-X(1)))
0061900       FN2=3.76*FO2
0062000       FH=S(3)*(1.0-X(1))
0062100       D11=DD1/453.6
```

```
0062200       T18=((D11*(-32200.2+13.95*(T5-77.))+X(2) *D11*(-104036.4          -
0062300      1+8.87*(T5-77.))+FCH*(-32200.2+13.95*(T16-77.))+S(7)*(-47548.8+7.4*-
0062400      2(T16-77.))+S(9)*(-169291.8+11.5 *(T16-77.))+FO2*(7.8 *(T16-77.))  -
0062500      3+FH2*(7.32*(T16-77.))+FH*(7.1*(T16-77.))+S(4) *(-104036.4          -
0062600      4+8.87*(T16-77.))-FCH*(-32200.2+13.95*(T5-77.))-FCO2*(-169291.8     -
0062700      5+11.5*(T5-77.))-FCO*(-47548.8+7.4*(T5-77.))-FH2O*(-104036.4        -
0062800      6+8.87*(T5-77.))-FH2*(7.1*(T5-77.)))+170177.3 *(S(7)+S(9)+FCH)      -
0062900      7+104719.39*(2.*FCH+S(4) +FH)+563.64*FN2+300.3 *FO2)                -
0063000      8/(11.5*( S(7)+S(9)+FCH)+8.87*(2.*FCH+S(4) +FH)+7.32*FN2+3.9 *FO2)
0063100       T18=(T18+460.)/1.8
0063200       H(8)=S(15)-T18
0063300       IF(DABS(H(8)).LT.0.2)  GO TO 556
0063400       IF(NC.NE.0) GO TO 401
0063500       X15=S(15)
0063600       TT18=T18
0063700       S(15)=T18
0063800       NC=1
0063900       GO TO 56
0064000   401 XX15=(X15*T18-TT18*S(15))/(X15-S(15)+T18-TT18)
0064100       X15=S(15)
0064200       TT18=T18
0064300       S(15)=XX15
0064400       GO TO 56
0064500   556 T=(T8-15.)*1.8
0064600       K2=DEXP(C11/T**3+C12/T**2+C13/T+C14)
0064700       A=FCO**2*(K2-1.)
0064800       B=-FCO**2*K2-FCO*FH2O*K2-FCO*FH2-FCO*FCO2
0064900       C=K2*FCO*FH2O-FCO2*FH2
0065000       XH=(-B-DSQRT(B**2-4.*A*C))/2./A
0065100       FCO9=FCO*(1.-XH)
0065200       FCO29=FCO2+FCO*XH
0065300       FH2O9=FH2O-FCO*XH
0065400       FH29=FH2+FCO*XH
0065500       H(9)=FCO9-S(10)
0065600       H(10)=FCO29-S(17)
0065700       H(11)=FH29-S(12)
0065800       H(12)=FH2O9-S(13)
0065900       DO 593 I=9,12
0066000       IF((DABS(H(I)).GT.0.02).AND.(I.EQ.9)) GO TO 594
0066100       IF((DABS(H(I)).GT.0.05).AND.(I.NE.9)) GO TO 594
0066200   593 CONTINUE
0066300       GO TO 559
0066400   594 S(10)=(FCO9+S(10))/2.
0066500       S(17)=(FCO29+S(17))/2.
0066600       S(12)=(FH29+S(12))/2.
0066700       S(13)=(FH2O9+S(13))/2.
0066800       GO TO 802
0066900   559 DO 5 I=1,7
0067000     5 DNSC(I)=0.
0067100       DNSC(2)=X(2)*DD3
0067200       DNSH(1)=453.6*FCH
0067300       DNSH(2)=0.
0067400       DNSH(3)=FCO9*453.6
0067500       DNSH(4)=FCO29*453.6
```

```
0067600      DNSH(5)=FH29*453.6
0067700      DNSH(6)=FH209*453.6
0067800      DNSH(7)=0.
0067900      CALL HEXCI(T10,DNSH,DNSC,T26,T11,T27,HA(9))
0068000      H(13)=S(16)-T27
0068100      IF(DABS(H(13)).LT.0.2)  GO TO 561
0068200      IF(NF.NE.0) GO TO 801
0068300      X16=S(16)
0068400      TT27=T27
0068500      S(16)=T27
0068600      NF=1
0068700      GO TO 802
0068800  801 XX16=(X16*T27- TT27 *S(16))/(X16-S(16)+T27-TT27)
0068900      X16=S(16)
0069000      TT27=T27
0069100      S(16)=XX16
0069200      GO TO 802
0069300  561 DO 6 I=1,7
0069400    6 DNSC(I)=0.
0069500      DNSC(2)=453.6*S(3)*X(1)
0069600      DNSC(7)=3.76*DNSC(2)
0069700      CALL HEXCI(T11,DNSH,DNSC,298.,T12,T31,HA(6))
0069800      T=(T12-15.)*1.8
0069900      K2=DEXP(C11/T**3+C12/T**2+C13/T+C14)
0070000      A=FC09**2*(K2-1.)
0070100      B=-FC09**2*K2-FC09*FH209*K2-FC09*FH29-FC09*FC029
0070200      C=K2*FC09*FH209-FC029*FH29
0070300      XL=(-B-DSQRT(B**2-4.*A*C))/2./A
0070400      F131=FCH
0070500      F132=0.
0070600      F133=FC09*(1.-XL)
0070700      F134=FC029+FC09*XL
0070800      F135=FH29+FC09*XL
0070900      F136=FH209-FC09*XL
0071000      F137=0.
0071100      F13=F131+F132+F133+F134+F135+F136+F137
0071200      H(14)=F133-S(7)
0071300      H(15)=F134-S(9)
0071400      H(16)=F135-S(3)
0071500      H(17)=F136-S(4)
0071600      DO 611 I=14,17
0071700      IF((DABS(H(I)).GT.0.05).AND.(I.NE.14)) GO TO 612
0071800      IF((DABS(H(I)).GT.0.002).AND.(I.EQ.14)) GO TO 612
0071900  611 CONTINUE
0072000      GO TO 565
0072100  612 S(7)=(S(7)+F133)/2.
0072200      S(9)=(S(9)+F134)/2.
0072300      S(3)=(S(3)+F135)/2.
0072400      S(4)=(S(4)+F136)/2.
0072500      GO TO 902
0072600  565 DNS1(1)=453.6*FCH
0072700      DNS1(2)=0.
0072800      DNS1(3)=453.6*F133
0072900      DNS1(4)=453.6*F134
```

```
0073000       DNS1(5)=453.6*F135*(1.-X(1))
0073100       DNS1(6)=453.6*F136
0073200       DNS1(7)=0.
0073300       DO 7 I=1,7
0073400     7 DNS2(I)=0.
0073500       DNS2(2)=907.2*(2.*FCH+0.5*F133+0.5*F135*(1.-X(1)))
0073600       DNS2(7)=3.76*DNS2(2)
0073700       CALL DMIX1(DNS1,DNS2,TOPFC,TAT,T15)
0073800       DO 8 I=1,6
0073900     8 DNSC(I)=DNS1(I)
0074000       DNSC(2)=DNS2(2)
0074100       DNSC(7)=DNS2(7)
0074200       DNSH(1)=0.
0074300       DNSH(2)=DNSC(2)/2.
0074400       DNSH(3)=0.
0074500       DNSH(4)=453.6*(FCH+F133+F134)
0074600       DNSH(5)=0.
0074700       DNSH(6)=453.6*(2.*FCH+F135*(1.-X(1)))
0074800       DNSH(7)=DNSC(7)
0074900       CALL HEXC1(T19,DNSH,DNSC,T15,T20,T16,HA(4))
0075000       H(18)=S(6) -T16
0075100       IF(DABS(H(18)).LT.0.2) GO TO 571
0075200       IF(NG.NE.0) GO TO 901
0075300       X6=S(6)
0075400       TT16=T16
0075500       S(6)= T16
0075600       NG=1
0075700       GO TO 902
0075800   901 XX6= (X6* T16- TT16 *S(6) )/(X6- S(6) +T16-TT16)
0075900       X6= S(6)
0076000       TT16=T16
0076100       S(6)= XX6
0076200       GO TO 902
0076300   571 CONTINUE
0076400       YH=F135/F13
0076500       YCO=F133/F13
0076600       G(1)= YCO-0.01
0076700       PH2=DSQRT(YH*(1.-X(1)) /(1.-YH*X(1)) *YH)
0076800       PCO=DSQRT(YCO**2/(1.-YH*X(1)))
0076900       EZ=1.26-0.00025*TOPFC
0077000       SR=0.44*DEXP(3650.*(1./TOPFC-1./450.))
0077100       C=0.2327*45.*PO2**.8*PH2O**0.4377*DEXP(-6652./TOPFC)
0077200       EX=11.85*0.0066*PCO*DEXP(9190.*(1./TOPFC-1./450.))
0077300       A=DLOG(PH2/PH2O*PO2**0.5)
0077400       C1=0.002385
0077500       D=4.31D-5*TOPFC
0077600       B=EZ+A*D
0077700       DA=2.*D
0077800       VOLT=B-DA*DLOG(.1  /C)-0.1  *SR-EX*(.1  /C1)
0077900       DC=CD*AREA  *VOLT*X(3) *NST
0078000       T5=(T5+460.)/1.8
0078100       G(2)= VOLT-0.75
0078200       G(3)= 0.5-VOLT
0078300       G(4)= VOLT*X(3) -300.
```

```
0078400       G(5)= 100.    -(-1.0148+DSQRT(1.0298-1.689D-3*(5.0976-DC/1000.)))/ -
0078500      18.444D-4
0078600       G(6)= T12-477.
0078700       G(7)= 398.-T12
0078800       G(8)= T31-422.
0078900       G(9)= 363.-T31
0079000       G(10)=0.6*F13-F135
0079100   171 IF(ISCALE .NE. 1) GO TO 8015
0079200 C*************************************************************************
0079300 C INSERT STATEMENTS DEFINING THE OBJECTIVE FUNCTION Y
0079400 C*************************************************************************
0079500 C      SAMPLE PROBLEM
0079600       TVV=VOLT*X(3)*NST
0079700       CALL RLIN1(13,CC1,CC2,TVV,CINV)
0079800       CINV=CINV*DC/1000.
0079900       Y=309.82*S(20)**0.63+1241.*(0.0526*(S(8)+S(12)))**.69+1820.*(0.05 -
0080000      1*(S(12)+S(3)))**0.69 +8.5136*X(3)*6                               -
0080020      2+1046.23*(HA(1)**0.6934+HA(2)**0.6934+HA(3)**0.6934+HA(4)**0.6934 -
0080040      3+HA(6)**0.6934+HA(9)**0.6934)                                     -
0080200      4+CINV
0080300   173 IF(KSCALE .NE. 0) GO TO 9901
0080400  8015 IF(ISCALE .NE. 3) GO TO 8020
0080500       X(KFR)=XXSAVE
0080600 C CALCULATE THE SUMS OF THE MAGNITUDES OF DG(I)/DX(KFR)--KFR=1,2,...,N--
0080700 C FORWARD DIFFERENCES, CALLING THE RESULTS PD(I)
0080800       DO 80 I=1,M
0080900       IF(KFR .NE. 1) GO TO 8003
0081000       PD(I)=0.0D00
0081100  8003 PDOLD=PD(I)
0081200       PD(I)=(G(I)-GB(I))/CDELTA
0081300       PD(I)=PDOLD+DABS(PD(I))
0081400   80  CONTINUE
0081500  8020 GO TO (8030,8022,8022,8030,9999,9901),K
0081600  8022 DO 8025 I=1,M
0081700  8025 G(I)=G(I)*SC(I)
0081800       ISCALE=2
0081900  8030 CONTINUE
0082000       IF(ISCALE .EQ. 3) GO TO 8040
0082100 C COMPUTE SUM OF VIOLATED CONSTRAINTS SQUARED; RESULT IS SIGMA0 IF G(I)'
0082200 C HAVE NOT BEEN SCALED, SIGMA1 IF THEY HAVE BEEN SCALED:
0082300  9901 SIGMA1=0.D00
0082400       IF(M .EQ. 0) GO TO 9905
0082500       IF(JEQ .EQ. 0) GO TO 9903
0082600       DO 9902 J=1,JEQ
0082700  9902 SIGMA1=SIGMA1+G(J)**2
0082800  9903 IF(JINEQ .EQ. 0) GO TO 9910
0082900       DO 9904 J=1,JINEQ
0083000       J1=JEQ+J
0083100       IF(G(J1) .LE. 0.) GO TO 9904
0083200       SIGMA1=SIGMA1+G(J1)**2
0083300  9904 CONTINUE
0083400  9910 IF(KSCALE .NE. 0) GO TO 9913
0083500       IF(K .NE. 6 .OR. ISCALE .NE. 1) GO TO 9911
0083600       SIGMA0=SIGMA1
```

```
0083700       DO 9912 I=1,M
0083800  9912 GB(I)=G(I)
0083900       GO TO 8022
0084000  9911 IF(ISCALE .EQ. 2) GO TO 9905
0084100       ISCALE=3
0084200       DO 8035 I=1,M
0084300  8035 GB(I)=G(I)
0084400 C RECLCULATE PD(I) FOR USE IN SUBROUTINE SCALE
0084500       GO TO 8000
0084600  8040 CONTINUE
0084700 C SCALE CONSTRAINTS IF AT BEGINNING OF NEW PENALTY LEVEL
0084800       CALL SCALE
0084900  7001 CONTINUE
0085000       GO TO 8022
0085100  9913 SIGMA0=SIGMA1
0085200       IF(K .NE. 6) GO TO 9905
0085300       DO 9914 I=1,M
0085400  9914 GB(I)=G(I)
0085500 C ERECT BARRIES AT BOUNDS OF INDEPENDENT VARIABLES
0085600  9905 SIGMA2=0.
0085700       IF((KBOUND .EQ. 0) .OR. (KCLIMB .EQ. 0)) GO TO 9909
0085800       IF(KBOUND .EQ. 2) GO TO 9907
0085900       DO 9906 I=1,N
0086000  9906 SIGMA2=SIGMA2+1.D00/(X(I)-XMIN(I))
0086100       GO TO 9909
0086200  9907 DO 9908 I=1,N
0086300  9908 SIGMA2=SIGMA2+1.D00/(X(I)-XMIN(I))+1.D00/(XMAX(I)-X(I))
0086400  9909 CONTINUE
0086500 C COMPUTE MODIFIED OBJECTIVE FUNCTION
0086600       Y=(1.-2.*KMIN)*Y
0086700       Z=Y-(PNLTY*SIGMA1+SIGMA2/(ALPHA*PNLTY**2))
0086800       IF(K .NE. 3) RETURN
0086900       IF(KMIN .EQ. 0) GO TO 9999
0087000       Y=-Y
0087100       Z=-Z
0087200  9999 CONTINUE
0087300       RETURN
0087400       END
0087500       SUBROUTINE PATTRN
0087600 C*************************************************************************
0087700 C THIS SUBROUTINE PERFORMS A PATTERN SEARCH FOR A MAXIMUM
0087800 C
0087900 C ALGORITHM OF HOOK AND JEEVES
0088000 C
0088100 C X1(N) IS THE EXPLORATORY POINT
0088200 C X2(N) IS THE NEW BASE POINT
0088300 C X3(N) IS THE OLD BASE POINT
0088400 C*************************************************************************
0088500       IMPLICIT REAL*8(A-H,O-Z)
0088600       DIMENSION EMAX(25),G(30),SC(30),X1(25),X2(25),X3(25),XK(25),      -
0088700      1          XMIN(25),XMAX(25)
0088800       COMMON DELTA,DX,EFAC,EPSLN,G,GRADX,PNLTY,SC,SIGMA0,SIGMA1,SIGMA2, -
0088900      1       TIME,X1,XMIN,XMAX,Y,Z
0089000       COMMON IGRAD,IOBJ,IOUT,IT,ITMAX,JEQ,JINEQ,KBOUND,KCLIMB,KMIN,M,N, -
```

```
     1      NTEST
  900 FORMAT('0    RESULTS OF EXPLORATORY MOVE'/,'0    ZSTAR=',D12.5,'  -
     1   ZBASE=',D12.5/)
  901 FORMAT('      X1(',I4,')=',D12.5,'     X2(',I4,')=',D12.5)                -
  902 FORMAT('0    RESULTS OF PATTERN-EXPLORATORY SEQUENCE'/,                   -
     1'0    ZBASE=',D12.5,8X,'ZSTAR=',D12.5,8X,'ZTEST=',D12.5/)
  903 FORMAT('      X3(',I4,')=',D12.5,5X,'X1(',I4,")=',D12.5,                  -
     15X,'X2(',I4,')=',D12.5)
  904 FORMAT(1X,4X,96('*')/)
C SET INITIAL VALUE FOR SEARCH INCREMENT
C SET INITIAL EXPLORATORY POINT AND INITIAL SEARCH DIRECTIONS
      IOBJ=0
      IT=0
      DO 1 I=1,N
      XK(I)=1.D+00
      EMAX(I)=1.D+01
      IF(KBOUND .EQ. 2) EMAX(I)=.01D+00*(XMAX(I)-XMIN(I))
      X2(I)=X1(I)
    1 X3(I)=X1(I)
      ZSTAR=Z
      YSTAR=Y
C INITIALIZE FLAG FOR EXPLORATORY SEQUENCE: LS=0 IF CONDUCTED ABOUT BASE
C     LS=1 IF CONDUCTED ABOUT PATTERN EXTRAPOLATION POINT
      LS=0
C EXPLORATORY SEQUENCE - ESTABLISH NEW BASE POINT
C X1 IS INITIAL POINT WITH VALUE ZSTAR;X2 NEXT BASE POINT WITH VALUE ZTE
  100 ZTEST=ZSTAR
      YTEST=YSTAR
      DO 104 I=1,N
  101 IF((X1(I).GE.XMAX(I)).AND.(XK(I).GT.0.)) GO TO 1010
      IF((X1(I).LE.XMIN(I)).AND.(XK(I).LT.0.)) GO TO 1010
      X2(I)=X1(I)+XK(I)*EMAX(I)
      IF(X2(I).GT.XMAX(I)) X2(I)=XMAX(I)
      IF(X2(I).LT.XMIN(I)) X2(I)=XMIN(I)
      CALL OBJECT(X2,Y1,Z1,2)
      IOBJ=IOBJ+1
      IF(Z1.LE.ZTEST) GO TO 1010
      XK(I)=XK(I)*2.D00
      IF(XK(I).GT.1.D+00) XK(I)=1.D+00
      IF(XK(I).LT.-1.D+00) XK(I)=-1.D+00
      GO TO 103
 1010 IF((X1(I).LE.XMIN(I)).AND.(XK(I).GT.0.)) GO TO 1011
      IF((X1(I).GE.XMAX(I)).AND.(XK(I).LT.0.)) GO TO 1011
 1020 X2(I)=X1(I)-XK(I)*EMAX(I)
      IF(X2(I).LT.XMIN(I)) X2(I)=XMIN(I)
      IF(X2(I).GT.XMAX(I)) X2(I)=XMAX(I)
      CALL OBJECT(X2,Y1,Z1,2)
      IOBJ=IOBJ+1
      IF(Z1.LE.ZTEST) GO TO 1011
      XK(I)=-XK(I)
      GO TO 103
C NO IMPROVEMENT AFTER EXPLORATORY SEQUENCE
 1011 X2(I)=X1(I)
      XK(I)=0.5D00*XK(I)
```

```
0094500       IF(XK(I).LT.0.) GO TO 1015
0094600       IF(XK(I).LT.0.5D+00*EFAC) XK(I)=0.5D+00*EFAC
0094700       GO TO 1018
0094800  1015 IF(DABS(XK(I)).LT.0.5D+00*EFAC) XK(I)=-0.5D+00*EFAC
0094900  1018 Z1=ZTEST
0095000       Y1=YTEST
0095100   103 ZTEST=Z1
0095200   104 YTEST=Y1
0095300 C END EXPLORATORY SEQUENCE
0095400 C BRANCH IF EXPLORATIRY HAS IMMEDIATELY FOLLOWED PATTERN EXTRAPOLATION
0095500       IF(LS.EQ.1) GO TO 350
0095600       ZBASE=ZTEST
0095700       YBASE=YTEST
0095800       IF(IOUT.LT.3) GO TO 106
0095900       IF(KMIN.EQ.0) GO TO 105
0096000       ZSTAR=-ZSTAR
0096100       ZBASE=-ZBASE
0096200   105 WRITE(6,900) ZSTAR,ZBASE
0096300       WRITE(6,901) ((I,X1(I),I,X2(I)),I=1,N)
0096400       IF(KMIN.EQ.0) GO TO 106
0096500       ZSTAR=-ZSTAR
0096600       ZBASE=-ZBASE
0096700   106 CONTINUE
0096800 C TEST FOR IMPROVEMENT FOLLOWING EXPLORATORY SEQUENCE
0096900       IF(ZBASE.GT.ZSTAR)GO TO 300
0097000 C TEST FOR COMPLETION OF PATTERN SEARCH
0097100       DO 201 I=1,N
0097200       IF(DABS(XK(I)).GT.0.5D+00*EFAC) GO TO 100
0097300   201 CONTINUE
0097400       GO TO 600
0097500 C PATTERN MOVE TO NEW EXPLORATORY POSITION
0097600   300 DO 301 I=1,N
0097700       X1(I)=X3(I)+2.D+00*(X2(I)-X3(I))
0097800       IF(X1(I).LT.XMIN(I)) X1(I)=XMIN(I)
0097900       IF(X1(I).GT.XMAX(I)) X1(I)=XMAX(I)
0098000       X3(I)=X2(I)
0098100   301 X2(I)=X1(I)
0098200       CALL OBJECT(X1,YSTAR,ZSTAR,2)
0098300       IOBJ=IOBJ+1
0098400       IT=IT+1
0098500       LS=1
0098600       GO TO 100
0098700   350 LS=0
0098800       IF(IOUT.LT.3) GO TO 303
0098900       IF(KMIN.EQ.0) GO TO 302
0099000       ZBASE=-ZBASE
0099100       ZSTAR=-ZSTAR
0099200       ZTEST=-ZTEST
0099300   302 WRITE(6,902) ZBASE,ZSTAR,ZTEST
0099400       WRITE(6,903)((I,X3(I),I,X1(I),I,X2(I)),I=1,N)
0099500       IF(KMIN.EQ.0) GO TO 303
0099600       ZBASE=-ZBASE
0099700       ZSTAR=-ZSTAR
0099800       ZTEST=-ZTEST
```

```
0099900   303 CONTINUE
0100000 C TEST FOR IMPROVEMENT FOLLOWING PATTERN MOVE
0100100   400 IF(ZTEST.LE.ZBASE) GO TO 500
0100200       ZBASE=ZTEST
0100300       YBASE=YTEST
0100400       GO TO 300
0100500 C INITIATE NEW EXPLORATORY SEQUENCE FROM OLD BASE POINT IF NECESSARY
0100600   500 DO 501 I=1,N
0100700       X1(I)=X3(I)
0100800   501 X2(I)=X1(I)
0100900       ZSTAR=ZBASE
0101000       YSTAR=YBASE
0101100       GO TO 100
0101200 C TERMINATE SEARCH
0101300   600 CALL OBJECT(X1,Y1,Z1,6)
0101400       IOBJ=IOBJ+1
0101500       Y=Y1
0101600       Z=Z1
0101700       IF(IOUT.EQ.3) WRITE(6,904)
0101800       RETURN
0101900       END
0102000       SUBROUTINE SCALE
0102100 C*************************************************************************
0102200 C THIS SUBROUTINE COMPUTES THE MULTIPLICATIVE SCALING CONSTANTS SC(I) SU
0102300 C THAT THE SUMS OF THE MAGNITUDES OF THE PARTIALS OF THE CONSTRAINTS ARE
0102400 C ROUGHLY EQUAL
0102500 C*************************************************************************
0102600       IMPLICIT REAL*8(A-H,O-Z)
0102700       DIMENSION G(30),GB(30),PD(30),SC(30),XDUMMY(25),XMIN(25),XMAX(25)
0102800       COMMON DELTA,DX,EFAC,EPSLN,G,GRADX,PNLTY,SC,SIGMA0,SIGMA1,SIGMA2, -
0102900      1NTEST
0103000       COMMON/DLK/PD,GB,SIGLPL,ALPHA,CFAC,KSCALE,IP,MPEN
0103100  2001 FORMAT('0',4X,96('*')/)
0103200  9000 FORMAT('0   CONSTRAINT SCALING INFORMATION FOR NEXT PENALTY LEVEL -
0103300     1'//)
0103400  9001 FORMAT('0    LARGEST MODIFIED SENSITIVITY=',D12.5/,                -
0103500     1'    UNSCALED SIGMA1=',D12.5/,                                     -
0103600     2'    SIGMA1 (SCALED) FROM LAST PENALTY LEVEL=',D12.5/,             -
0103700     3'    TEMPORARY (NOT NORMALIZED) SIGMA1=',D12.5/,                   -
0103800     4'    SCADJT (SCALING FACTOR NORMALIZATION CONSTANT)=',D12.5,//)
0103900  9002 FORMAT('    GB(',I4,')=',D12.5,10X,'PD(',I4,')=',D12.5,10X,       -
0104000     1'SC(', I4,')=',D12.5)
0104100 C TEST FOR PROBLEMS NOT REQUIRING CONSTRAINT SCALING
0104200       IF(M .LE. 1) GO TO 9050
0104300 C CALCULATE LOWER BOUNDS OF SENSITIVITIES AND FIND LARGEST IN MAGNITUDE
0104400       IF(JEQ .EQ. 0) GO TO 9006
0104500       DO 9005 I=1,JEQ
0104600       AP=PD(I)
0104700       IF(AP .LT. 1.0D00) AP=1.0D00
0104800       AG=DABS(GB(I))
0104900       IF(AG .LT. 1.0D00) AG=1.0D00
0105000  9005 SC(I)=AP*AG
0105100  9006 IF(JINEQ .EQ. 0) GO TO 9008
0105200       DO 9007 I=1,JINEQ
```

```
0105300       I1=JEQ+I
0105400       AP=PD(I1)
0105500       IF(AP .LT. 1.0D00) AP=1.0D00
0105600       AG   =GB(I1)
0105700       IF(AG .LT. 1.0D00) AG=1.0D00
0105800  9007 SC(I1)=AP*AG
0105900  9008 SCBIG=0.0D00
0106000       DO 9009 I=1,M
0106100       SCBIG=DMAX1(SCBIG,SC(I))
0106200  9009 CONTINUE
0106300       DO 9010 I=1,M
0106400       SC(I)=DSQRT(SCBIG/SC(I))
0106500  9010 G(I)=SC(I)*GB(I)
0106600       SIGMA0=SIGMA1
0106700 C COMPUTE TEMPORARY SIGMA1 FOR USE IN NORMALIZATION
0106800       TSIG1=0.D00
0106900       IF(JEQ .EQ. 0) GO TO 9015
0107000       DO 9012 J=1,JEQ
0107100  9012 TSIG1=TSIG1+G(J)**2
0107200  9015 IF (JINEQ .EQ. 0) GO TO 9025
0107300       DO 9020 J=1,JINEQ
0107400       J1=JEQ+J
0107500       IF (G(J1) .LE. 0.0D00) GO TO 9020
0107600       TSIG1=TSIG1+G(J1)**2
0107700  9020 CONTINUE
0107800 C COMPUTE NORMALIZATION FACTOR FOR SCALING FACTORS--SCADJT
0107900  9025 IF(TSIG1 .NE. 0.D00) GO TO 9030
0108000       SCADJT=1.0D00
0108100       GO TO 9040
0108200  9030 IF(IP .GT. 1) GO TO 9035
0108300       SCADJT=DSQRT(SIGMA0/TSIG1)
0108400       GO TO 9040
0108500  9035 SCADJT=DSQRT(SIGLPL/TSIG1)
0108600  9040 IF(SCADJT .EQ. 0.0D00) SCADJT=1.D00
0108700       DO 9045 I=1,M
0108800       G(I)=GB(I)
0108900       SC(I)=SC(I)*SCADJT
0109000  9045 CONTINUE
0109100       SIGMA1=SIGMA0
0109200       IF(IOUT .LE. 2) GO TO 9050
0109300 C WRITE SCALING INFORMATION
0109400       WRITE(6,9000)
0109500       WRITE(6,9001) SCBIG,SIGMA0,SIGLPL,TSIG1,SCADJT
0109600       WRITE(6,9002) ((I,G(I),I,PD(I),I,SC(I)),I=1,M)
0109700       WRITE(6,2001)
0109800  9050 RETURN
0109900       END
0110000       SUBROUTINE COEFF(MJL)
0110100 C*****************************************************************
0110200 C THIS SUBROUTINE COMPUTES AN INITIAL VALUE FOR THE PENALTY COEFFICIENT
0110300 C*****************************************************************
0110400       IMPLICIT REAL*8(A-H,O-Z)
0110500       DIMENSION G(30),G1(30),GB(30),PD(30),SC(30),X(25),XMIN(25),      -
0110600      1          XMAX(25)
```

```
      COMMON DELTA,DX,EFAC,EPSLN,G,GRADX,PNLTY,SC,SIGMA0,SIGMA1,SIGMA2, -
     1       TIME,X,XMIN,XMAX,Y,Z
      COMMON IGRAD,IOBJ,IOUT,IT,ITMAX,JEQ,JINEQ,KBGUND,KCLIMB,KMIN,M,N, -
     1       NTEST
      COMMON/DLK/PD,GB,SIGLPL,ALPHA,CFAC,KSCALE,IP,MPEN
      MJL1=0
      XJONES=1.-2.*KMIN
      IF(MJL .EQ. 1) GO TO 500
      IF(SIGMA0*SIGMA2 .EQ.0.) GO TO 200
C MIXED PENALTY CASE
  100 MJL1=1
      GO TO 300
  101 PNLTY1=PNLTY
      PNLTY=1.D+00
      GO TO 400
  102 ALPHA=(PNLTY/PNLTY1)**2
      PNLTY=PNLTY1
      RETURN
  200 IF(SIGMA2 .GT. 0.) GO TO 400
C PENALTY FUNCTION COMPOSED OF FUNCTIONAL CONSTRAINTS-BOUNDS ARE NOT
C CONSIDERED
  300 GRADY=0.
      GRADG=0.
      GSUM=0.
      CALL OBJECT(X,Y,Z1,2)
      DO 310 J=1,M
  310 GSUM=GSUM+G(J)*G(J)
      YMINUS=Y
      DO 301 J=1,M
  301 G1(J)=G(J)
      DO 305 I=1,N
      U=X(I)
      IF(DABS(U) .LE. 1.D+00) GO TO 302
      X(I)=(1.D+00+DELTA)*U
      CALL OBJECT(X,Y1,Z1,2)
      YPLUS=Y1
      GRADY=GRADY+((YPLUS-YMINUS)/(DELTA*U))**2
      V=0.
      DO 303 J=1,M
  303 V=V+(G(J)+G1(J))*(G(J)-G1(J))/(DELTA*U)
      GRADG=GRADG+V*V
      GO TO 305
  302 X(I)=U+DELTA
      CALL OBJECT(X,Y1,Z1,2)
      YPLUS=Y1
      GRADY=GRADY+((YPLUS-YMINUS)/DELTA)**2
      V=0.
      DO 304 J=1,M
  304 V=V+(G(J)+G1(J))*(G(J)-G1(J))/DELTA
      GRADG=GRADG+V*V
  305 X(I)=U
      IF(GRADG .GE. 1.D-02) GO TO 306
      PNLTY=CFAC*DSQRT(GRADY)
  307 IF((GRADY.LE.DELTA).AND.(GSUM.GE.DELTA)) PNLTY=CFAC*DABS(Y)/GSUM
```

```
0116100       IF((GSUM.LT.DELTA).AND.(DABS(Y).GT.DELTA)) PNLTY=DABS(Y)*CFAC
0116200       IF((GRADY.LE.DELTA).AND.(DABS(Y).LE.DELTA)) PNLTY=1.D+00
0116300       IF(MJL1.EQ.1) GO TO 101
0116400       RETURN
0116500   306 PNLTY=DSQRT(GRADY/GRADG)*CFAC
0116600       GO TO 307
0116700 C PENALTY FUNCTION COMPOSED OF BOUNDS-FUNCTIONAL CONSTRAINTS ARE NOT
0116800 C     PRESENT: CHOOSE THE PENALTY COEFFICIENT SUCH THAT THE SQUARE OF
0116900 C     GRAD(Z) IS MINIMIZED
0117000   400 XNUM=0.
0117100       ALPHA=1.D+00
0117200       XDENOM=0.
0117300       YMINUS=Y
0117400       SUM=SIGMA2
0117500       DO 403 I=1,N
0117600       U=X(I)
0117700       IF(DABS(U).LE.1.D+00) GO TO 401
0117800       X(I)=(1.D+00+DELTA)*U
0117900       CALL OBJECT(X,Y1,Z1,2)
0118000       YPLUS=Y1
0118100       GO TO 402
0118200   401 X(I)=U+DELTA
0118300       CALL OBJECT(X,Y1,Z1,2)
0118400       YPLUS=Y1
0118500       DYDX=(YPLUS-YMINUS)/DELTA
0118600   402 X(I)=U
0118700       U=1.D+00/(XMAX(I)-X(I))**2-1.D+00/(X(I)-XMIN(I))**2
0118800       XNUM=XNUM+U**2
0118900   403 XDENOM=XDENOM+DYDX*U
0119000       PSQ=XJONES*XNUM/XDENOM
0119100       IF(Y .EQ. 0.) GO TO 404
0119200       PSQMIN=1.D-02*SUM/DABS(Y)
0119300       IF(PSQ .LT. PSQMIN) PSQ=PSQMIN
0119400       PSQMAX=1.D+02*SUM/DABS(Y)
0119500       IF(PSQ .GT. PSQMAX) PSQ=PSQMAX
0119600   404 PNLTY=DSQRT(PSQ)
0119700       IF(PNLTY .EQ. 0.) PNLTY=1.D+00
0119800       IF(MJL1 .EQ. 1) GO TO 102
0119900   500 RETURN
0120000       END
0120100       SUBROUTINE HEXC1(THI,DNSH,DNSC,TCI,THO,TCO,       HA)
0120200 C*************************************************************************
0120300 C THIS SUBROUTINE IS TO CAL. THE ENERGY ANALYSIS FOR HAET EXCHANGER
0120400 C*************************************************************************
0120500       IMPLICIT REAL*8(A-H,O-Z)
0120600       DIMENSION GS(7),HS(7),HCAS(7),HCBS(7),HCCS(7)
0120700       DIMENSION DNSH(7),DNSC(7)
0120800       COMMON /ETHDA/ GS,HS,HCAS,HCBS,HCCS
0120900       COMMON /U/ U
0121000       COMMON /CN/ CN
0121100 C ASSUME THE MEAN TEMP. AT COLD AND HOT SIDE
0121200       THM=(THI+10.+TCI)/2.
0121300       TCM=(THI-10.+TCI)/2.
0121400 C CAL. CC AND CH
```

```
0121500      21 CC=0.
0121600         CH=0.
0121700         DO 1 IA=1,7
0121800         IF(DNSC(IA).EQ.0.) GO TO 2
0121900         CC=CC+DNSC(IA)*(HCAS(IA)+HCBS(IA)*TCM+HCCS(IA)*TCM**2)
0122000       2 IF(DNSH(IA).EQ.0.) GO TO 1
0122100         CH=CH+DNSH(IA)*(HCAS(IA)+HCBS(IA)*THM+HCCS(IA)*THM**2)
0122200       1 CONTINUE
0122300C CHOOSE THE CMAX. ,CMIN.
0122400         IF(CC.GT.CH) GO TO 3
0122500         CMAX=CH
0122600         CMIN=CC
0122700         GO TO 4
0122800       3 CMAX=CC
0122900         CMIN=CH
0123000       4 CONTINUE
0123100         HA=CH*CMIN/U
0123200         UA=HA*U
0123300C CAL. THE HEAT EXCHANGER EFFECTIVENESS
0123400         HE=(1.-DEXP((-UA/CMIN)*(1.-CMIN/CMAX)))/(1.-(CMIN/CMAX)*DEXP(( -
0123500        1-UA/CMIN)*(1.-CMIN/CMAX))))
0123600         THO=THI-HE*(CMIN/CH)*(THI-TCI)
0123700         TCO=HE*(CMIN/CC)*(THI-TCI)+TCI
0123800         QT=HE*CMIN*(THI-TCI)
0123900         IF((DABS((THO+THI)/2.-THM).LT.(DABS((THO+THI)/2.+THM)*0.005)  -
0124000        1).AND.(DABS((TCO+TCI)/2.-TCM).LT.(DABS((TCO+TCI)/2.+TCM)*0.005))) -
0124100        2 GO TO 11
0124200         THM=(THO+THI)/2.
0124300         TCM=(TCI+TCO)/2.
0124400         GO TO 21
0124500      11 THM=(THO+THI)/2.
0124600         TCM=(TCO+TCI)/2.
0124700         RETURN
0124800         END
0124900C*********************************************************************
0125000C THIS SUBROUTINE IS TO CAL. THE BALANCE AROUND THE MIXER
0125100C*********************************************************************
0125200         IMPLICIT REAL*8(A-H,O-Z)
0125300         DIMENSION GS(7),HS(7),HCAS(7),HCBS(7),HCCS(7)
0125400         DIMENSION DNS1(7),DNS2(7),DNS(7)
0125500         COMMON /ETHDA/ GS,HS,HCAS,HCBS,HCCS
0125600C CAL. THE TOTAL THERMAL CONST.
0125700         TCAS1=0.
0125800         TCBS1=0.
0125900         TCCS1=0.
0126000         TCAS2=0.
0126100         TCBS2=0.
0126200         TCCS2=0.
0126300         DO 1 IA=1,7
0126400         TCAS1=TCAS1+DNS1(IA)*HCAS(IA)
0126500         TCAS2=TCAS2+DNS2(IA)*HCAS(IA)
0126600         TCBS1=TCBS1+DNS1(IA)*HCBS(IA)
0126700         TCBS2=TCBS2+DNS2(IA)*HCBS(IA)
```

```
      TCCS1=TCCS1+DNS1(IA)*HCCS(IA)
      TCCS2=TCCS2+DNS2(IA)*HCCS(IA)
    1 CONTINUE
C ASSUME THE INITIAL VALUE
      TOUT=(TIN1+TIN2)/2.
C CAL. THE ENERGY BALANCE
    2 TOUTC=(TCAS1*TIN1+TCAS2*(TIN2-TOUT)+TCBS1/2.*(TIN1**2-TOUT**2)     -
     1+TCBS2/2.*(TIN2**2-TOUT**2)+TCCS1*(TIN1**3-TOUT**3)/3.+TCCS2       -
     2*(TIN2**3-TOUT**3)/3.)/TCAS1
      CALL CONV1(TOUT,TOUTC,1,NC)
      GO TO (3,2),NC
    3 CONTINUE
      RETURN
      END
      SUBROUTINE RLIN1(N,XT,YT,X,ANS)
C THIS SUBROUTINE IS TO CAL. LINEAR INTERPOLATION.
C THE ALGORITHM REQUIRES XT VECTOR TO BE IN ASCENDING ORDER......
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  XT(20),YT(20)
      I=2
      IF(X.LE.XT(1)) GO TO 20
      I=N
      IF(X.GE.XT(N)) GO TO 20
      DO 10 I=2,N
      IF(X.LE.XT(I)) GO TO 20
   10 CONTINUE
   20 ANS=YT(I-1)+(YT(I)-YT(I-1))/(XT(I)-XT(I-1))*(X-XT(I-1))
      RETURN
      END
      SUBROUTINE CONV1(XV,YV,NR,NC)
C*****************************************************************************
C THIS SUBROUTINE IS USING WEGSTEIN METHODE TO SOLVE NONLINEAR EQUATION WITH   *
C ONE VARIABLE                                                                 *
C*****************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XA(2),YA(2)
      IF(DABS((XV-YV)/(XV+YV)).LT.0.001) GO TO 6
      IF(NC.LE.1) GO TO 5
      XT=(XA(NR)*YV-YA(NR)*XV)/(XA(NR)-XV+YV-YA(NR))
      XA(NR)=XV
      YA(NR)=YV
      XV=XT
      RETURN
    5 XA(NR)=XV
      YA(NR)=YV
      XV=YV
      NC=2
      RETURN
    6 XV=YV
      NC=1
      RETURN
      END
```